



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

**“DESARROLLO DEL SISTEMA DE REQUISICIONES PARA LA
EMPRESA HIDROELECTRICA ABANICO S.A. APLICANDO EL
ENTORNO DE PROGRAMACION NODE.JS”**

TRABAJO DE TITULACIÓN: PROYECTO TÉCNICO

Para optar el Grado Académico de:

INGENIERA EN SISTEMAS INFORMÁTICOS

AUTORA: DIANA ELIZABETH GÓMEZ GARCÍA

TUTOR: Dr. JULIO SANTILLAN CASTILLO Msg.

Riobamba – Ecuador

2018

©2018, Diana Elizabeth Gómez García

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

El Tribunal del trabajo de titulación certifica que: El proyecto técnico “DESARROLLO DEL SISTEMA DE REQUISICIONES PARA LA EMPRESA HIDROELÉCTRICA ABANICO. S.A. APLICANDO EL ENTORNO DE PROGRAMACIÓN NODE.JS”, de responsabilidad de la señora DIANA ELIZABETH GÓMEZ GARCÍA, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, quedando autorizada su presentación.

Dr. Julio Santillán Mgs.

VICEDECANO DE LA FACULTAD DE

INFORMÁTICA Y ELECTRÓNICA _____

Ing. Patricio Moreno Mgs.

DIRECTOR DE LA ESCUELA DE

INGENIERÍA EN SISTEMAS _____

Dr. Julio Santillán Mgs.

DIRECTOR DEL TRABAJO DE _____

TITULACIÓN

Ing. Jonny Guaiña Mgs.

MIEMBRO DEL TRIBUNAL _____

Yo, DIANA ELIZABETH GÓMEZ GARCÍA soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de titulación; y, el patrimonio intelectual del mismo pertenece a la Escuela Superior Politécnica de Chimborazo.

Diana Elizabeth Gómez García

DEDICATORIA

Dedico este trabajo en primer lugar a mis queridos padres, que han sido importantes pilares en mi vida, gracias por siempre demostrarme su cariño y ayuda incondicional sin importar las diferencias de opinión que hemos tenido.

A mi hermana por siempre estar allí para mí, por su apoyo en mi carrera estudiantil y por los consejos que me ha brindado a lo largo de mi vida.

A mi esposo Ángel, a mi eterno compañero, gracias por brindarme su amor incondicional; a mis hijas Dayanna, Daniela y Danna por ser mi inspiración para mejorar cada día.

Mi triunfo es el de ustedes, ¡Los quiero!, Gracias

Diana

AGRADECIMIENTO

Gracias a mi esposo Angel, mi alma gemela, por ser en todo momento mi apoyo incondicional y mi felicidad encaja en una sola persona.

Debo agradecer de manera especial y sincera al Dr. Julio Santillán, porque gracias a su dirección se culminó este trabajo; gracias por siempre atenderme con amabilidad y cordialidad en cada revisión de mi trabajo de titulación.

Diana

TABLA DE CONTENIDO

INDICE DE TABLAS	ix
INDICE DE FIGURAS	x
INDICE DE GRÁFICOS	xi
ÍNDICE DE ANEXOS	xii
ÍNDICE DE ABREVIATURAS	xiii
RESUMEN	xiv
ABSTRACT	xv
INTRODUCCIÓN	16
CAPITULO I	
1. MARCO TEÓRICO	23
1.1 Requisiciones de compra	23
1.1.1 Característica	24
1.1.2 Flujo de la requisición	24
1.2 HERRAMIENTAS APLICACIÓN WEB.....	25
1.2.1 NodeJS	26
1.2.2 Django.....	30
1.3 METODOLOGÍA DE DESARROLLO.....	37
1.3.1 Metodología MSF	37
1.3.2 Objetivos MSF	37
1.3.3 Principios Fundamentales.....	38
1.3.4 Fases	38
1.4 ESTANDAR ISO-9126	39
1.4.1 Métricas y Sub-métricas	39
1.4.2 Usabilidad	41
1.4.3 Escala de Usabilidad del Sistema (System Usability Scale, SUS)	41
CAPITULO II	
2. MARCO METODOLÓGICO	43
2.1. Visión y Alcance	43
2.1.1. Definición del Problema.....	43
2.1.2 Visión del Proyecto	44
2.1.3 Perfiles de Usuario	44
2.1.4 Ámbito del Proyecto	45
2.1.5 Herramientas a Utilizar	45
2.1.6 Objetivos de Proyecto.....	46

2.1.7	<i>Análisis y Gestión del riesgo</i>	47
2.1.8	<i>Planificación Inicial</i>	54
2.1.9	<i>Estudio de Factibilidad</i>	64
2.2	<i>Planificación</i>	66
2.2.1	<i>Definición y Especificación de Requerimientos</i>	66
2.2.2	<i>Diseño Lógico</i>	78
2.2.3	<i>Diseño Físico</i>	86
2.3	<i>Desarrollo</i>	87
2.3.1	<i>Nomenclatura y Estándares</i>	87
2.3.2	<i>Recolección y análisis de evidencias primitivas</i>	88
2.3.3	<i>Modelo</i>	88
2.3.4	<i>Vista</i>	89
2.3.5	<i>Controlador</i>	90
2.4	<i>Estabilización</i>	90
2.4.1	<i>Revisión del Sistema</i>	90
2.4.2	<i>Plan de pruebas</i>	92
2.5	<i>Soporte</i>	93
2.5.1	<i>Manuales de Usuario</i>	93

CAPITULO III

3	MARCO DE RESULTADOS	94
3.1	Problema	94
3.2	Población	94
3.3	Análisis Test SUS	95
3.4	Análisis de resultados	97

CONCLUSIONES	99
---------------------------	----

RECOMENDACIONES	100
------------------------------	-----

BIBLIOGRAFÍA

GLOSARIO

ANEXOS

INDICE DE TABLAS

Tabla 1-1: Métricas y Sub-métricas	40
Tabla 2-2: Herramientas.....	46
Tabla 3-2: Riesgos	47
Tabla 4-2: Impacto Técnico	48
Tabla 5-2: Probabilidad.....	48
Tabla 6-2: Exposición al riesgo.....	48
Tabla 7-2: Código de colores	49
Tabla 8-2: Análisis del riesgo	49
Tabla 9-2: Línea de corte	50
Tabla 10-2: Riesgo 1	51
Tabla 11-2: Riesgo 3	52
Tabla 12-2: Riesgo 5	53
Tabla 13-2: Equipo de Trabajo	54
Tabla 14-2: Creación de cuenta.....	58
Tabla 15-2: Relleno de la requisición de compra.....	58
Tabla 16-2: Autorización de las requisición de compra.....	58
Tabla 17-2: Consulta de reportes	59
Tabla 18-2: Generación de reportes	59
Tabla 19-2: Registro de Nuevos Usuarios	59
Tabla 20-2: Presupuesto Proyecto.....	66
Tabla 21-2: Requisitos Servidor.....	69
Tabla 22-2: Requisitos Cliente.....	70
Tabla 23-2: Interfaces de Software	70
Tabla 27-2: Requerimiento #4 - Caso de uso - Gestión de Notificaciones	75
Tabla 29-2: Herramientas.....	78
Tabla 30-2: Extensiones Archivos	88
Tabla 31-2: Componente del sistema evaluado.....	92
Tabla 32-2: Equipos para Pruebas.....	93
Tabla 33-3: Tabulación datos Test SUS.....	95

INDICE DE FIGURAS

Figura 1-1: Flujo de la requisición.....	25
Figura 2-1: Creación servidor websocket	30
Figura 3-1: Puerto localhost.....	30
Figura 4-1: Modelo.py	31
Figura 5-1: Urls.py.....	31
Figura 6-1: Views.py	32
Figura 7-1: Sintaxis IF	35
Figura 8-1: Sintaxis FOR.....	36
Figura 9-1: Sintaxis FOR range()	36
Figura 10-1: Sintaxis Clases	36
Figura 11-2: Orgánico Funcional – Cantón Morona.....	56
Figura 12-2: Nomenclatura de flujogramas	60
Figura 13-2: Creación de la cuenta	61
Figura 14-2: Relleno de la requisición.....	61
Figura 15-2: Autorización de la requisición	62
Figura 16-2: Consulta de reportes.....	62
Figura 17-2: Generación de reportes.....	63
Figura 18-2: Registro de Nuevo Usuario	63
Figura 19-2: Requerimiento # 4 - Caso de Uso - Gestión de Notificaciones.....	75
Figura 20-2: Arquitectura interna	76
Figura 21-2: Arquitectura externa.....	76
Figura 22-2: Arquitectura externa.....	77
Figura 23-2: Arquitectura detallada	77
Figura 24-2: Gestión de usuarios - Diagrama de Secuencia Requerimiento #1.....	79
Figura 25-2: Gestión de inventario - Diagrama de secuencia Requerimiento #2	79
Figura 26-2: Gestión de requisiciones - Diagrama de secuencia Requerimiento #3	80
Figura 27-2: Gestión de notificaciones - Diagrama de secuencia Requerimiento #4	80
Figura 28-2: Reportes - Diagrama de secuencia Requerimiento #5	81
Figura 29-2: Diagrama de clases.....	82
Figura 30-2: Diagrama de objetos – pedido de requisición	83
Figura 31-2: Autenticación de usuarios.....	84
Figura 32-2: Cuenta Administrativo (Secretaria-Gerente).....	84
Figura 33-2: Cuenta operativa (Operador).....	85
Figura 34-2: Reportes	85
Figura 35-2: Diagrama de implementación	86
Figura 36-2: Modelo físico base de datos	87
Figura 37-2: models.py - Sistema de requisiciones	88
Figura 38-2: Templates	89
Figura 39-2: Urls.py	89
Figura 40-2: Views.py	90
Figura 41-2: Código fuente - Servidor de notificaciones.....	91

INDICE DE GRÁFICOS

Gráfico 1-3: Evaluación por pregunta.....	96
Gráfico 2-3: Usabilidad Sistema de Requisición	97

ÍNDICE DE ANEXOS

ANEXO A: ENCUESTA RECOPIACIÓN DE INFORMACIÓN

ANEXO B: DIAGRAMA GANTT

ANEXO C: DIAGRAMA DE CASOS DE USO

ANEXO D: MANUAL DE USUARIO

ANEXO E: TEST SUS

ÍNDICE DE ABREVIATURAS

MSF	Microsoft Solutions Framework
UML	Lenguaje de Modelado Unificado
CASE	Ingeniería de Software Asistida por Computadora
MVC	Model View Controller
MTV	Model Template View
SQL	Structured Query Language
URL	Uniform Resource Locator
ISO	International Organization for Standardization
DBMS	Data Base Managment System
SUS	System Usability Scale
API	Application Programming Interface

RESUMEN

La presente investigación tuvo como finalidad el desarrollo de un sistema de requisiciones para la empresa Hidroeléctrica Abanico. S.A. aplicando el entorno de programación NodeJS. El desarrollo del sistema se lo hizo a través de Microsoft Solutions Framework, esta metodología fue útil para la gestión de riesgos, ayuda con el soporte y mantenimiento, además se enfocó en la gestión y administración del proyecto para lograr el impacto deseado. El sistema se diseñó para automatizar el proceso de pedidos de requisición a través de una aplicación web que se encuentre activa en un browser en las 3 principales instalaciones de la empresa (oficina, casa de máquinas, represa). En el desarrollo se integró Django para el módulo de administración, este framework ayudo en el desarrollo rápido y pragmático. El servidor de notificaciones fue realizado en NodeJS, utilizando websockets, permitiendo la comunicación bidireccional en tiempo real entre cliente y el servidor en un mismo socket, utilizando una conexión TCP. La medición de resultados se lo hizo por medio del sistema de escalas de usabilidad (SUS) lo que permitió medir la usabilidad por medio de encuestas a los usuarios obteniendo una aceptación satisfactoria del sistema. De esta forma se pudo concluir que la utilización del sistema ayuda en el seguimiento y control de requisiciones. Al ser una aplicación escalable, se recomienda la implementación del módulo de proveedores para complementar el pedido de la requisición.

PALABRAS CLAVE: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <INGENIERÍA DE SOFTWARE>, <PROGRAMACIÓN DE APLICACIÓN WEB>, <OPTIMIZACIÓN DE PROCESOS INDUSTRIALES>, <NODEJS (FRAMEWORK)>, <JAVASCRIPT (SOFTWARE)>, <DJANGO (FRAMEWORK)>, <REQUISICIONES>.

ABSTRACT

The purpose of the present investigation was to develop a requisition system for the hydroelectric company Abanico S.A. flushing the NodeJS programming environment. The development of the system was done through Microsoft Solutions Framework, this methodology was useful for risk management, help with support and maintenance, also focused on the management and administration of the project to achieve the desired impact. The system was designed to automate the requisition request process through a web application that is active in a browser in the three main facilities of the company (office, powerhouse, and dam). In the development Django was integrated for the administration module, this framework helped in the rapid and pragmatic development. The notification server was realized in NodeJS, using websockets, allowing bidirectional communication in real time between the client and the server in the same socket, using a TCP connection. The measurement of the results was done through the system of scales for usability (SUS), which allowed to measure the usability by means of user surveys, obtaining a satisfactory acceptance of the system. In this way it was possible to conclude that the use of the system helps in the follow-up and control of requisitions. Being a scalable application, it is recommended to implement the suppliers module to complement the requisition request.

Key words: <TECHNOLOGY AND ENGINEERING SCIENCES>, <SOFTWARE ENGINEERING>, <WEB APPLICATION PROGRAMMING>, <INDUSTRIAL PROCESS OPTIMIZATION>, <NODEJS (FRAMEWORK)>, <JAVASCRIPT (SOFTWARE)>, <DJANGO (FRAMEWORK)>, <REQUIREMENTS>

INTRODUCCIÓN

La tecnología está cambiando al mundo que habitamos, excluyendo barreras de tiempo y distancia, permitiendo compartir información. En la actualidad permite generar herramientas de apoyo en la solución de problemas o necesidades que requiera el ser humano. El internet junto con la tecnología sea convertido en el gran aliado de los usuarios.

El internet se insertó en la sociedad cambiando el concepto de comunicaciones, permitiendo estar conectados desde cualquier lugar del planeta; basándose en un computador, un browser y conexión a internet. Los usuarios pueden utilizar aplicaciones web subidas a la red, para solucionar alguna necesidad o problema existente.

El mundo del desarrollo web avanza rápido, cada día surgen nuevas herramientas y nuevas tendencias que debemos implementar si queremos seguir siendo competitivos dentro del mercado. El presente de las aplicaciones es la respuesta inmediata a cualquier interacción, lo que en tecnología se conoce como Tiempo Real. (García, 2014, p.2).

Las aplicaciones en tiempo real son más bien aplicaciones bidireccionales, lo cual quiere decir que a diferencia de tecnologías como *Ajax*, que únicamente envían datos del cliente al servidor y reciben una respuesta, la tecnología bidireccional puede enviar datos del cliente al servidor y de igual manera del servidor al cliente, mostrando la información en vivo que se encuentre en el servidor; todo esto es posible con node.js utilizando Socket.IO. (García, 2014, p.3).

La presente investigación se basa en el desarrollo del sistema de requisiciones para la empresa hidroeléctrica Abanico. S.A. aplicando el entorno de programación node.js. La requisición es el pedido de compra de algún bien o servicio de un departamento para la liberación de recursos y la adquisición del mismo. El entorno NodeJS permitirá la comunicación en tiempo real a través del lenguaje Javascript para la creación de notificaciones entre el personal administrativo y operativo.

Se integrará el framework Django a través del lenguaje Python, para la administración de la aplicación, este framework también comprende las interacciones cliente-servidor en los sitios web y la programación del lado del servidor. La aplicación se la realizará en base a la metodología MSF por sus siglas en inglés *Microsoft Solutions Framework* cuyas fases comprenden visión y alcance, planificación, desarrollo, estabilización, instalación y soporte.

La aplicación tendrá como principal funcionalidad la comunicación en tiempo real a través de notificaciones entre el operador y secretaria mediante una interfaz web. Dicha aplicación ayudará con la creación de notificaciones de un pedido de requisición entre el personal operativo (operador) y el personal administrativo (secretaria).

El presente trabajo de titulación consta de 3 capítulos que se describen a continuación:

El capítulo 1 comprende el marco teórico donde se recogerá la información para el desarrollo del sistema, como conceptos generales, definiciones, características, herramientas utilizadas.

El capítulo II se enfocará al desarrollo de la aplicación con su ingeniería, aplicando la metodología de desarrollo de software MSF (*Microsoft Solutions Framework*).

El capítulo III se detallará el análisis de los resultados que se obtendrá en base al test SUS (*System Usability Scale*) que evalúa la usabilidad del producto final.

ANTECEDENTES

En el 2003 se creó la compañía Hidroabanico S.A. para construir proyectos de generación y autogeneración hidroeléctrica, centrales que generan electricidad amigable con la naturaleza y el medio ambiente. El proyecto hidroeléctrico Abanico localizado en la región sur oriental del Ecuador, en la provincia de Morona Santiago. Las obras de toma de agua localizadas a 5 y 15 km de la ciudad de Macas.

Hidroabanico basa su gestión en la responsabilidad social empresarial, que es la nueva forma de gestión de las empresas para operar de forma sustentable en lo económico, social y ambiental. Esta central proporciona empleo a habitantes de la zona por medio de turno rotativos. Además de facilitar el transporte hacia las centrales, la empresa dota de lo necesario para realizar exitosamente lo encomendado a cada empleado a este procedimiento se lo conoce como requisiciones.

La dinámica del entorno empresarial en la actualidad obliga a las empresas a mirar constantemente las maneras de mejorar sus diferentes procedimientos con el fin de mantenerse competitivas y rentables, si bien la correcta interconexión de todas las áreas de una empresa es crucial para que esta funcione eficientemente y logre una mejora continua. Así mismo, la tecnología actual permite insertar diferentes escenarios y visualizar los posibles resultados que se obtendrían bajo ciertos parámetros establecidos. (Hernández, 2015, p.4)

Las empresas implementan modificaciones en su actuar diario, los resultados de insertar la tecnología permiten a los directivos tomar decisiones sobre la dirección que desean seguir dentro de un panorama más definido. Actualmente la mayoría de empresas cuentan con problemas de cultura organizacional que reflejan fallas en los sistemas que implementan, específicamente en la parte de cumplimiento de procesos.

El proceso de notificaciones de requisición, es uno de los procesos importantes para la correcta ejecución de proyectos, ya que el abastecimiento oportuno de materiales impacta directamente en los tiempos de ejecución, es importante conocer su funcionamiento y las opciones de mejora que puedan implementarse. La ineficiencia en los procesos de requisición se ve reflejado a través de la lentitud, exceso de papeleo. Este proceso abarca desde que se solicita hasta que se realiza la orden de requisición.

Una requisición de compra (documento) es originada por un empleado en la empresa (planta 1 o planta 2) y enviada a través del transporte hacia la oficina principal ubicada en la ciudad de Macas.

Cuando el documento se encuentra en la oficina recibe la autorización por parte del gerente. Una vez autorizada, la secretaria procede al despacho hacia el lugar donde se originó la requisición con el fin de abastecer bienes o servicios requeridos por los trabajadores. Cabe recalcar que el proceso de notificar las requisiciones toma de 1 a 2 días, tampoco existe ninguna revisión de lo solicitado antes de la autorización por lo cual consume tiempo y recursos ocasionando deficiencia en el control de las requisiciones realizadas.

FORMULACIÓN DEL PROBLEMA

En la actualidad la empresa Hidroeléctrica Abanico S.A. no cuenta con un sistema informático para el control y seguimiento de los pedidos de requisición. Actualmente se conoce que la empresa lleva el proceso de forma manual lo cual genera retardo en la notificación y entrega de la necesidad originada en la oficina, casa de máquinas o represa. Razón por la cual es recomendable desarrollar un sistema web que ayude con esta labor.

SISTEMATIZACIÓN DEL PROBLEMA

¿Se cuenta con una base de datos de las requisiciones realizadas por los empleados de la central?

¿Se cuenta con una base de datos de las requisiciones autorizadas por la secretaria en la oficina principal?

¿La implementación de un sistema de requisiciones en la empresa Hidroeléctrica Abanico S.A. ayudará al control adecuado de las requisiciones?

JUSTIFICACIÓN DEL TRABAJO DE GRADO

JUSTIFICACIÓN TEÓRICA

A medida que los frameworks de Aplicaciones Web, evolucionan, se requiere obtener mejores ventajas y beneficios en el entorno de programación y de esa manera poder disminuir el tiempo de desarrollo y facilidad para la construcción de Aplicaciones Web.

Las plataformas más populares compiten ferozmente entre ellas por ofrecer mejores funcionalidades y prestaciones. Pero al final acabaremos eligiendo una plataforma que se adapte adecuadamente al usuario y sus requerimientos.

Las aplicaciones web son síncronas por naturaleza. El usuario interactúa con la interfaz web que se muestra en el navegador, el navegador realiza peticiones de vuelta al servidor de acuerdo a la interacción del usuario, y el servidor responde con HTML. Esto representa un problema ya que el usuario no podrá conocer los cambios que surjan en el servidor a menos que realice una nueva petición.

La Web Asíncrona es fundamentalmente diferente, y es esta diferencia la que revoluciona el comportamiento de las aplicaciones web. En la Web Asíncrona es posible entregar al usuario cambios espontáneos en la presentación, ya que ahora podemos mantener una representación exacta del sistema en el navegador del usuario.

Una de las principales metas al plantear el desarrollo de la aplicación web es lograr brindar accesibilidad inmediata a la información generada.

NodeJS gestiona todas sus tareas de una forma Asíncrona, esta arquitectura orientada a eventos consta de muchos módulos útiles que trabajan con un único hilo de ejecución que es el encargado de organizar todo el flujo de trabajo que se deberá realizar. NodeJS a través de los websockets ayudará al sistema de requisiciones a mostrar notificaciones en “tiempo real.”

Por otro lado, se utilizará el framework Django con el lenguaje programación Python el cual provee estructuras de datos eficientes y de alto nivel, además de un enfoque simple pero efectivo a la programación orientada a objetos. La sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de este un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. Proporciona un equilibrio entre lo práctico y lo conceptual. Python viene con una gran biblioteca de módulos que se pueden usar para hacer toda clase de tareas que abarcan desde programación para web a gráficos.

JUSTIFICACIÓN APLICATIVA

La empresa no cuenta con un eficiente control y manejo de las requisiciones creadas según necesidades que se crean en la planta 1 y planta 2, peticiones que son llevadas en forma escrita a través del transporte una vez terminado el turno en la tarde. Estas peticiones llegan a manos de la Secretaria quien, como ente encargado de autorizar, permite la compra de los bienes que se solicita. Sin saber de antemano si ya existe en las plantas tal requisición, lo que hace que este proceso se vuelva ineficiente, obsoleto y tardío.

La presente investigación pretende solucionar dicho problema al implementar un control a las requisiciones realizadas, que brinden seguridad, confiabilidad y se ajusten a las necesidades de la empresa Hidroeléctrica Abanico S.A.

La utilización del sistema informático facilitará la gestión de las requisiciones por parte de los empleados en las plantas y el control de las mismas por parte de la secretaria en la oficina principal. A futuro se contará con una solución informática que adaptará el proceso de creación, control y autorización de las requisiciones de compra en la empresa HidroAbanico S.A.

La requisición será generada por un empleado de la planta a través del sistema web que la almacenará periódicamente a través de un ID, esto será reportado inmediatamente a la Secretaria quien consultará en la base de datos si dicha petición ya fue generada con anterioridad para su cancelación o proceder con la autorización para su respectiva compra.

La aplicación actuará como intermediario entre el empleado (generador) y la secretaria (autorizadora) para dar seguimiento a las requisiciones realizadas por la central.

La aplicación a desarrollarse constará de los siguientes módulos:

- Módulo Administración
- Módulo Requisiciones
- Módulo Reportes

OBJETIVOS

OBJETIVO GENERAL

Desarrollar el sistema de requisiciones para la empresa Hidroeléctrica Abanico. S.A. aplicando el entorno de programación NODE.JS

OBJETIVOS ESPECIFICOS

- Analizar la gestión de requisiciones desde una perspectiva que se ajuste al diseño de una aplicación web.
- Diseñar una aplicación web utilizando las herramientas informáticas Python y Javascript del lado del servidor utilizando NODE.JS.
- Implementar la aplicación web mediante el uso de notificaciones en tiempo real para la empresa HidroAbanico S.A.
- Evaluar la usabilidad del sistema de requisiciones mediante la aplicación del test SUS a los empleados de la empresa HidroAbanico S.A.

CAPITULO I

1. MARCO TEÓRICO

En este capítulo comprende la recopilación de información para el conocimiento del tema requisiciones como concepto general, características y flujo que sigue la requisición al momento de ser generadas hasta su despacho. Además, resumir información de los framework NodeJS y Django; y sus lenguajes de programación respectivamente *Java Script* y *Python* para la utilización en el desarrollo de la aplicación. Además, la metodología MSF (*Microsoft Solutions Framework*), al ser una metodología tradicional está diseñada para grupos pequeños de desarrollo. Finalmente, se aborda la norma ISO 9126 para la evaluar la usabilidad del producto software.

1.1 Requisiciones de compra

En la actualidad las empresas utilizan la requisición como el concepto de la adquisición de algún recurso necesario para el uso en la empresa. La requisición implica un proceso desde que surge la necesidad hasta que está sea despachada para abastecer la necesidad existente.

La requisición de compra es un proceso fundamental en toda empresa ya que permite el adecuado funcionamiento de los procesos, consiste en una solicitud en la cual se detalla la materia prima, productos, herramientas, suministros o cualquier tipo de necesidad de algún departamento o unidad. Si existe en bodega lo requerido se procede con el despacho y si no se procede con la compra de lo solicitado. La requisición es importante por cuanto permite dar seguimiento y control a los gastos que generan la ejecución de los procesos en la empresa. (Talavera, 2015, p.136).

El acto de comprar es uno de los sucesos más antiguos de la humanidad, desde el tiempo del trueque, donde nace el concepto de vender o comprar y el término requisición que se refiere al verbo “requiere”, es decir, la necesidad o el requisito de algo, una requisición de compra en el área industrial es el acto de obtener el producto o servicio adecuado, con la calidad correcta, al precio justo, en el tiempo preciso y con el proveedor adecuado. La requisición de compra, es un documento interno de la empresa. (Talavera, 2015, p.137).

Para el caso específico de la empresa Hidroeléctrica Abanico SA la requisición es realizada por los operarios de la planta, quienes cumplen varias funciones como, supervisión, monitoreo, mantenimiento, control, limpieza, seguridad y transporte desde las diferentes estaciones que forman la planta hidroeléctrica. Para cada una de las funciones de los operarios se necesita contar con insumos, suministros, materiales o herramientas que son provistos por el departamento administrativo en donde se despacha o aprueba la compra, para luego ser enviado al lugar de origen de la requisición en el transporte de la empresa.

1.1.1 *Característica*

Una requisición de compra es la forma de notificación en la que cualquier departamento de una empresa informa la necesidad de comprar algún recurso. En este documento se debe incluir el departamento que lo requiere, datos de los artículos a comprar, cantidad de artículos solicitados, detalle de los artículos, fecha de generación de la requisición entre otros datos importantes que lo requiera la empresa. (Alliende, 2015, p.49).

Una requisición no es una orden de compra, por lo cual no puede ser utilizado para la compra de bienes o servicios, ni autorización de pago hacia el proveedor; la requisición sirve para que los administrativos de la empresa suplan las necesidades en cada uno de los procesos en ejecución, así como también para poder: (Alliende, 2015, p.54)

- Programar, dirigir y controlar las actividades de recepción, despacho y envío de los requerimientos de cada departamento.
- Verificar el uso de los artículos entregados.
- Mantener un registro y control del stock de los artículos disponibles en la bodega.
- Generar reportes de las requisiciones solicitadas, aceptadas, despachadas y recibidas.
- Planificar futuras compras.

1.1.2 *Flujo de la requisición*

La requisición de compra, es un documento que admite la necesidad de comprar dicho bien, máquina, servicio o algún objeto de la requisición. Generada esta requisición de compra a nivel

interno también se genera una solicitud de adquisición que es enviada al departamento encargado de la liberación de recursos, para su posterior aprobación o rechazo del pedido. (Alliende, 2015, p.56).

En la figura 1-1 muestra el camino que sigue la requisición desde su generación en algún departamento de la empresa hasta su posterior despacho.

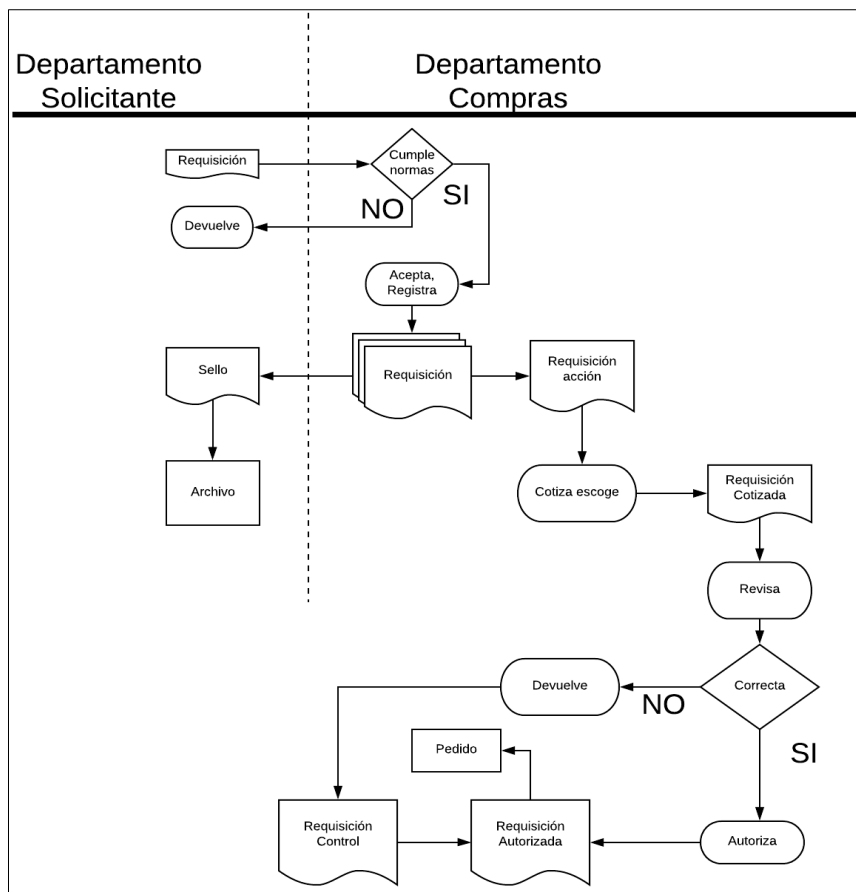


Figura 1-1: Flujo de la requisición

Realizado por: Gómez, Diana, 2017.

1.2 HERRAMIENTAS APLICACIÓN WEB

El mundo del desarrollo web avanza rápido, cada día surgen nuevas herramientas y nuevas tendencias que debemos implementar si se quiere seguir siendo competitivos. Existen varias tecnologías que se pueden emplear para programar aplicaciones web. En esta investigación se realizará un sistema web de notificación de requisiciones en tiempo real para esto se utilizará el

lenguaje de programación multiplataforma Python para la administración del sitio del lado del servidor, este lenguaje de tipado dinámico y sintaxis limpia se centra en la lógica del programa. Para notificar en tiempo real se utilizará NodeJS con javascript de lado del servidor por medio de websockets.

1.2.1 *NodeJS*

Node.js es una plataforma desarrollada en tiempo de ejecución *JavaScript* de *Chrome* para construir fácilmente aplicaciones de red escalables. *Node.js* usa un modelo de operaciones Entada/Salida sin bloqueo y orientado a eventos, que lo hace liviano y eficiente, perfecto para datos intensivos en tiempo real aplicaciones que se ejecutan en dispositivos distribuidos. El ecosistema de paquetes de *Node.js*, *npm*, es el más grande de librerías de código abierto en el mundo. (Node.js Foundation, 2018)

NodeJS es un entorno en tiempo de ejecución multiplataforma, de código abierto, que permite trabajar con *javascript* en la capa del servidor, con arquitectura orientada a eventos y basado en el motor V8 de Google, este motor está diseñado para correr en el browser y ejecutar el código *javascript* de una manera rápida. Fue creado con el enfoque de ser útil en la creación de aplicaciones web altamente escalables. (Flores, 2016, p 28)

NodeJS usa un hilo de ejecución (*thread*) para organizar el flujo de trabajo. Las tareas son gestionadas de forma asíncrona por este *thread*. Un *poll* de *threads* constituye librerías con un entorno multithread asíncrono. Finalizado el trabajo de los *threads*, se lanza un evento receptado por NodeJS. (Castrelo, 2014, p. 13)

Uno de los principales motivos al escoger *Node.js*, es por su utilización para aplicaciones *realtime*, ya que necesitan una conexión persistente entre el cliente y el servidor. Es especialmente útil cuando se van a realizar muchas operaciones simultáneas, sobretodo operaciones I/O.

1.2.1.1 *Aplicaciones con NodeJS*

Podemos generar desde pequeños módulos, hasta aplicaciones completas, ya que *node.js* permite manejar cientos de miles de conexiones concurrentes; así mismo por la manera en que está implementado, es totalmente adecuado para lo que tiene que ver con intercomunicación a gran escala, como aplicaciones de mensajería, juegos multijugador, hasta el “*cloud computing*” o “computación en la nube” y las aplicaciones en tiempo real, el cual es el tema de énfasis en esta ocasión. (García, 2014, p 3).

1.2.1.2 *Javascript*

En el diseño web, uno de los componentes más significativos es la posibilidad de obtener sitios dinámicos, que se distingan y atraigan la atención de sus visitantes. Los usuarios buscan páginas navegables, intuitivas y con calidad, tanto en el diseño como en los contenidos. Para conseguir este objetivo es inevitable utilizar un lenguaje que acreciente las funcionalidades del estático código *html*. *Javascript* es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas que permitan interactuar con el navegador de manera eficaz, proporcionando a las páginas web dinamismo e interactividad. (Flores, 2016, p 11)

Javascript fue desarrollado por *Netscape* para el uso en millones de páginas web y aplicaciones de servidor, es un lenguaje de programación dinámico que soporta programación orientada a objetos basada en prototipos, es similar a *java* y *C++* en cuanto a sintaxis, ya que los bucles y condicionales funcionan de igual forma. *Javascript* permite la construcción de objetos en tiempo de ejecución, así como también variables, funciones, *scripts* dinámicos, recuperación de código y listas de variables. (Mozilla MDN, 2015).

Existen varias implementaciones de JavaScript entre las más destacadas podemos mencionar a *SpiderMonkey* (*Netscape*), *Rhino* (*Norris Boyd*), *TraceMonkey* (*Firefox 3.5*), *JagerMonkey* (*Firefox 4*), *IonMonkey* (*Firefox 5*), *V8* (*Google Chrome*), *JavaScriptCore* (*Safari Apple*), *Carakan* (*Opera*), *Chakra* (*Internet Explorer*). El entorno más utilizado de *Javascript* son los navegadores web, también es utilizado como lenguaje interpretado de lado del servidor ya que mediante un API se puede manejar peticiones y respuestas http, así como generar páginas web de manera dinámica.

La idea es hacer disponible poderosas funciones a través de técnicas de programación sencillas y estándares, expandiendo el alcance del lenguaje y facilitando la creación de programas útiles para la web. (Gauchat, 2012, p.87).

Otra aplicación común de *javascript* es como lenguaje interpretado de lado del servidor (web). Un servidor web escrito en *javascript* podría exponer objetos host que representen objetos de una petición y una respuesta *http*, los cuales podrían ser manipulados por un programa en *javascript* para generar páginas web de manera dinámica. (Mozilla MDN, 2015).

1.2.1.3 *Aplicaciones en tiempo real*

Las aplicaciones en tiempo real son más bien aplicaciones bidireccionales, lo cual quiere decir que a diferencia de tecnologías como *AJAX*, que únicamente envían datos del cliente al servidor y reciben una respuesta, la tecnología bidireccional puede enviar datos del cliente al servidor y de igual manera del servidor al cliente, mostrando la información en vivo que se encuentre en el servidor; todo esto es posible con *node.js* utilizando *Socket.IO*. (García, 2014, p 3)

1.2.1.4 *Websockets*

Websocket prácticamente es mantener una conexión abierta entre un servidor web y un navegador, lo cual permite que tanto el navegador como el servidor envíen datos cuando lo deseen. Como la conexión es persistente, el intercambio de datos es muy rápido y se le llama “tiempo real”. (García, 2014, p 4)

Websocket es una tecnología que proporciona un canal de comunicación *full-dúplex* (bidireccional) sobre un único *socket* TCP. Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse en cualquier aplicación cliente/servidor. Como las conexiones TCP ordinarias sobre puertos distintos al 80 normalmente son bloqueadas por los administradores de redes, el uso de esta tecnología proporciona una solución a este tipo de limitaciones, ya que provee una funcionalidad similar a la apertura de varias conexiones en distintos puertos, pero multiplexando diferentes servicios *websocket* sobre un único puerto TCP. (Castrelo, 2014, p 17)

Los *websockets* es una tecnología basada en el protocolo TCP que permite la comunicación bidireccional sobre un solo *socket*, utilizado en aplicaciones cliente/servidor. A través, de una API por sus siglas en inglés *Application Programming Interface*.

1.2.1.5 *Socket.IO*

Nace la librería para *node.js* llamada *Socket.IO*, una librería *javascript* que nos permite manejar eventos en tiempo real mediante una conexión TCP y nos provee de todas las funcionalidades necesarias. (García, 2014, p 4).

Socket.IO permite la comunicación bidireccional basada en eventos en tiempo real. Funciona en todas las plataformas, navegadores o dispositivos, centrándose por igual en la fiabilidad y la velocidad. *Socket.IO* está construido sobre la API *websockets* (lado del cliente) y *Node.js*. Es una de las bibliotecas más dependientes de *npm* (node package manager). (Socket.io, 2018)

1.2.1.5.1 *Características*

- Análisis en tiempo real: Envía datos a los clientes que se representan como contadores, gráficos o registros en tiempo real. (Socket.io, 2018)
- Transmisión binaria: Comenzando en 1.0, es posible enviar cualquier de ida y vuelta: imagen, audio, video. (Socket.io, 2018)
- Mensajería instantánea y chat: El "Hola mundo" de Socket.IO es una aplicación de chat en unas pocas líneas de código. (Socket.io, 2018)
- Colaboración documental: Permite a los usuarios editar al mismo tiempo un documento y ver los cambios de los demás. (Socket.io, 2018)

1.2.1.6 *Creación Servidor*

La creación de un servidor de *websockets*, desarrollado en *Node.js* por medio de la librería *Socket.io* para facilitar la comunicación a través de un único *socket* entre el cliente y el servidor. Para implementar dicho servidor, se crea un paquete de Node con *npm init* –y para generar un *package.json*.

En el fichero *server/main.js* se incluirá las librerías *express* y *socket.io*, que se instalan vía *npm*: *\$ npm install –save express* y *\$ npm install –save socket.io*. Se crea la aplicación con *express*, que

se pasa a un servidor *http* y se conecta al servidor *sockets.io*. El servidor escuchara desde el local host por el puerto 8080.

```
var express = require('express'),
    app      = express();

var server = require('http').createServer(app)
    io      = require('socket.io').listen(server);
```

Figura 2-1: Creación servidor websocket
Realizado por: Gómez, Diana, 2017.

```
server.listen(8080, function() {
    console.log('Servidor corriendo en _____')
});
```

Figura 3-1: Puerto localhost
Realizado por: Gómez, Diana, 2017.

1.2.2 *Django*

Django es un *framework* web de alto nivel escrito en *python* que permite el desarrollo de sitios web rápido, seguro y mantenibles. Desarrollado por programadores experimentados, *Django* se encarga de gran parte de las complicaciones del desarrollo web. Es gratuito, de código abierto y tiene comunidades de colaboración extendidas por todo el mundo que se encargan de actualizar día a día el *framework* y la documentación (Mozilla MDN, 2018)

Este *framework* se basa en el principio de *Don't Repeat Yourself* es decir que lo que ya se ha inventado no se tiene que volver a programar. Cuenta con una amplia gama de componentes que evita volver a programarlos, por ejemplo, la autenticación de usuarios está compuesta por métodos para registrarse, iniciar sesión, cerrar sesión, así como un panel de administración para el sitio a desarrollar, cuenta con formularios, plantillas, reportes, entre otros. *Django* utiliza la arquitectura MVC por sus siglas Modelo – Vista – Controlador.

La capa Modelo cuenta con el archivo *models.py* para estructurar y manipular los datos de la aplicación web. (Mozilla MDN, 2018).

Contiene la representación de las tablas de la base de datos en forma de clases, todo esto se lo hace utilizando código *python*, esto permite que mediante el mapeador de objeto-relacional sea fácil la migración a cualesquiera de los *dbms* soportados por *python*, ya que no se necesita escribir sentencias SQL. A continuación, se muestra el ejemplo de un modelo.

mysite/news/models.py

```
from django.db import models
class Reporter(models.Model):
    full_name = models.CharField(max_length = 70)

    def __str__(self):
        return self.full_name

class Article(models.Model):
    pub_date = models.DateField( )
    headline = models.CharField(max_length=200)
    content = models.TextField( )
    reporter = models.ForeignKey(Reporter, on_delete=models.CASCADE)

    def __str__(self):
        return self.headline
```

Figura 4-1: Modelo.py

Realizado por: Gómez, Diana, 2017.

La capa Vista en *Django* “Plantilla” cuenta con el archivo *urls.py* para encapsular la lógica responsable de procesar una petición del usuario y devolver la respuesta. Se define los URL de acceso y la vista que esta servirá. Utiliza la sintaxis de expresiones regulares. (Mozilla MDN, 2018)

mysite/news/urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('articles/<int:year>/', views.year_archive),
    path('articles/<int:year>/<int:month>/', views.month_archive),
    path('articles/<int:year>/<int:month>/<int:pk>/',
        views.article_detail),
]
```

Figura 5-1: Urls.py

Realizado por: Gómez, Diana, 2017.

La capa Controlador en *Django* “Views”, cuenta con dos tipos de vistas las que se basan en clases y aquellas definidas por funciones. El archivo *views.py* provee la sintaxis amigable para los diseñadores que permite mostrar la información que se presentará al usuario. (Mozilla MDN, 2018).

```
mysite/news/views.py

from django.shortcuts import render

from .models import Article

def year_archive (request, year):

    a_list = Article.objects.filter(pub_date__year = year)

    context = {'year':year, 'article_list': a_list}

    return render(request, 'news/year_archive.html', context)
```

Figura 6-1: Views.py

Realizado por: Gómez, Diana, 2017.

Base de Datos: Psycopg2 es la librería que se debe instalar en Python para que exista comunicación con el DBMS de PostgreSQL.

1.2.2.1 Características

- **Completo:** Provee lo que los desarrolladores quisieran hacer, sigue principios de diseño consistentes. (Mozilla MDN, 2018).
- **Versátil:** Puede construir casi cualquier tipo de sitio web, desde sistemas manejadores de contenidos y wikis, hasta redes sociales y sitios nuevos. Funciona con cualquier framework cliente-servidor, y puede devolver contenido en casi cualquier formato (incluyendo HTML, RSS feeds, JSON, XML, etc). (Mozilla MDN, 2018).
- **Seguro:** Django ayuda a los desarrolladores a evitar errores comunes de seguridad. Por ejemplo: Proporciona una manera segura de administrar cuentas de usuario y contraseñas, evitando así errores comunes como colocar la información de la sesión en *cookies* donde es vulnerable (en lugar de eso las *cookies* solo contienen una clave y los datos se

almacenan en la base de datos) o se almacenan directamente las contraseñas en un *hash* de contraseñas (longitud fija creada al enviar la contraseña). (Mozilla MDN, 2018).

- Escalable: cada parte de la arquitectura es independiente de las otras, Teniendo en cuenta una clara separación entre las diferentes partes significa que puede escalar para aumentar el tráfico al agregar hardware en cualquier nivel: servidores de cache, servidores de bases de datos o servidores de aplicación. Sitios más concurridos han escalado a Django para satisfacer sus demandas. Por ejemplo, Instagram. (Mozilla MDN, 2018).
- Mantenible: Está escrito usando principios y patrones de diseño para fomentar la creación de código mantenible y reutilizable. En particular, utiliza el principio No te repitas "*Don't Repeat Yourself*" para que no exista una duplicación innecesaria, reduciendo la cantidad de código. Promueve la agrupación de la funcionalidad relacionada en "aplicaciones" reutilizables y en un nivel más bajo, agrupa código relacionado en módulos. (Mozilla MDN, 2018).
- Portable: Está escrito en Python, el cual se ejecuta en muchas plataformas. Lo que significa que no está sujeto a ninguna plataforma en particular, y puede ejecutar sus aplicaciones en muchas distribuciones de Linux, Windows y Mac OS X. Además, Django cuenta con el respaldo de muchos proveedores de alojamiento web, y que a menudo proporcionan una infraestructura específica y documentación para el alojamiento de sitios de Django. (Mozilla MDN, 2018).

1.2.2.2 Python

Python es un lenguaje de programación fácil de aprender y potente. Tiene eficaces estructuras de datos de alto nivel y una solución de programación orientada a objetos simple pero eficaz. La elegante sintaxis de *python*, su gestión de tipos dinámica y su naturaleza interpretada hacen de él el lenguaje ideal para scripts y desarrollo rápido de aplicaciones, en muchas áreas y en la mayoría de las plataformas. (Rossum, 2015, p. 1)

Un lenguaje interpretado o de *script* es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora. (González, 2017, p.7).

La ventaja de los lenguajes compilados es que su ejecución es más rápida. Sin embargo, los lenguajes interpretados son más flexibles y más portables, *python* tiene, no obstante, muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi-interpretado. En *python*, como en *java* y muchos otros lenguajes, el código fuente se traduce a un pseudo código máquina intermedio llamado *bytecode* la primera vez que se ejecuta, generando archivos.pyc o .pyo (bytecode optimizado), que son los que se ejecutarán en sucesivas ocasiones. (Gonzalez, 2017, p.7).

Python al ser un lenguaje de alto nivel facilita la estructuración de aplicaciones complejas ya que se pueden integrar archivos por lotes o *scripts* realizados en *Unix*, incorpora diccionarios de datos, así como también arreglos de tamaño flexible.

Si comparamos la longitud de un programa escrito en java o C++ los escritos en *python* son mucho más cortos debido a las siguientes razones: (Rossum, 2015, p. 7)

- En una sola instrucción se puede expresar tipos de datos de alto nivel para operaciones complejas.
- El agrupamiento de sentencias se realiza mediante sangría (*indentación*) en lugar de *begin/end* o llaves.
- Las variables o argumentos se las declara o inicializa el momento que son requeridas no al inicio del programa.

1.2.2.2.1 Tipos de Datos

En *python* podemos destacar tres tipos de datos importantes numéricos, cadenas de caracteres y listas, pese a que no es necesario definir una variable y asignarle un tipado.

Números

Python puede representar números enteros, reales y complejos. Los números enteros pueden ser positivos o negativos sin decimales, se los puede representar con el tipo *int* o el tipo *long*, lo único que los difiere es que el tipo *long* permite almacenar números más grandes, se debe tener precaución en el uso de este tipo de dato y solo utilizarlo si es estrictamente necesario. Los números reales tienen decimales en *python* se representan con el tipo *float* y también con el tipo decimal dependerá de la longitud del número a almacenar, se los expresa con la parte entera seguido de un punto y por último la parte decimal. Los números complejos tienen parte imaginaria

y es muy improbable su utilización se representan con el tipo de dato `complex`. (Rossum, 2015, p. 13)

Cadenas de Caracteres

Las cadenas de caracteres son texto que pueden estar encerradas en comillas simples ('...') o en comillas dobles ("..."). El símbolo `\` se puede utilizar para realizar el escapado de comillas sencillas o simples. También se puede usar las comillas simples dentro de comillas dobles y viceversa para que puedan ser omitidas. Las cadenas también admiten operadores como `+`, que funciona realizando una concatenación de las cadenas utilizadas como operandos y `*`, en la que se repite la cadena tantas veces como lo indique el número utilizado como segundo operando. (Rossum, 2015, p. 16)

Listas

La lista es un tipo de colección ordenada. Equivalente a lo que en otros lenguajes se conoce por arrays, o vectores. Las listas pueden contener cualquier tipo de dato: números, cadenas, *booleanos*, ... y también listas. Crear una lista es tan sencillo como indicar entre corchetes, y separados por comas, los valores que queremos incluir en la lista. (Rossum, 2015, p. 20)

Control de Flujo

Python maneja el control de flujo de los programas mediante la utilización de condicionales y bucles. Los programas no son una lista de órdenes secuenciales por lo que se necesita del uso de instrucciones condicionales que controlen la fluidez y orden de ejecución del programa. La sentencia *if* es la forma más simple de condición. Python utiliza la sentencia *elif* como una abreviación *else if*. (Rossum, 2015, p. 25)

```
X = 1

if x > 0:
    print('número positivo')
elif x < 0:
    print('número negativo')
else:
    print('cero')
```

Número positivo

Figura 7-1: Sintaxis IF

Realizado por: Gómez, Diana, 2017.

Sentencia *for... in*

El ciclo *for.... in* se utiliza para iterar sobre una secuencia y su uso es más común en el uso de listas. (Rossum, 2015, p. 25)

```
colors = ['white','black','red','blue','yellow']
for color in colors:
    print(color)

white
black
red
blue
yellow
```

Figura 8-1: Sintaxis FOR

Realizado por: Gómez, Diana, 2017.

La iteración de un *for* para que realice la iteración con condiciones de inicio y fin del formato se lo hace utilizando al función *range()*. (Rossum, 2015, p. 26)

```
for i in range(3):
    print(i)

0
1
2
```

Figura 9-1: Sintaxis FOR range()

Realizado por: Gómez, Diana, 2017.

Orientación a Objetos: Clases

Las clases representan objetos. Las definiciones de clases, como las definiciones de funciones deben ejecutarse para tener efecto. En *python*, podemos definir e instanciar una clase de la siguiente forma. (Rossum, 2015, p. 65).

```
Class Ojo:
    def __init__(self, forma,color, tamaño):
```

Figura 10-1: Sintaxis Clases

1.3 METODOLOGÍA DE DESARROLLO

El diseño y desarrollo de software exige el cumplimiento de requerimientos especificados, pero cumpliendo con los parámetros de una metodología probada y garantizada solo así se podrá tener un producto de calidad que satisfaga las necesidades del cliente y cumpla con parámetros de calidad de software. Para el desarrollo del sistema se eligió la metodología MSF en sus siglas en ingles *Microsoft Solutions Framework*, esta metodología totalmente practica y personalizable para entregar soluciones tecnológicas con menos personas en el desarrollo, menos riesgos, con resultados de calidad y de impacto comercial.

1.3.1 Metodología MSF

Microsoft Solutions Framework puede ser una herramienta eficaz para las organizaciones que desean desarrollar de manera rápida soluciones tecnológicas de alta calidad y relevantes para el negocio. Su flexibilidad permite adaptarlo de manera sencilla a la mayoría de proyectos tecnológicos, lo que ayuda a los equipos a comunicarse y coordinar las actividades más importantes. (Microsoft, 2017).

MSF es una metodología que ayuda con menos recursos a la realización de proyectos tecnológicos de alta calidad. Esta metodología alinea los objetivos de negocio y de tecnología, establece los objetivos, roles y responsabilidades, implementa proceso iterativo, gestiona riesgos y responde ante cambios. Toda solución tecnológica se divide en cinco fases principales:

- a) Visión y alcance
- b) Planificación
- c) Desarrollo
- d) Estabilización
- e) Implantación

1.3.2 Objetivos MSF

Los objetivos de MSF son: (Landázuri, 2015, p.1)

- Alinear objetivos empresariales y tecnológicos.
- Trazar correctamente los roles, estableciendo puntos de control.
- Establecer puntos iterativos, estableciendo puntos de control.
- Establecimiento oportuno de riesgos.
- Respuestas oportunas y efectivas a cambios no esperados.

1.3.3 *Principios Fundamentales*

Entre los principios fundamentales de MSF se tiene: (Landázuri, 2015, p.1)

- Comunicación abierta entre los miembros del equipo.
- Trabajo hacia una visión compartida y un objetivo común.
- Delegación de responsabilidades claras y compartidas entre los responsables con sus permisos necesarios.
- Adaptabilidad y apertura a cambios con soluciones óptimas y oportunas.
- Principios de calidad de software con resultados auditables.

1.3.4 *Fases*

1.3.4.1 *Visión y Alcances*

La fase Visión y Alcances está dada por la visión clara y conjunta del equipo para lograr satisfacer al cliente y motivar al equipo. Se define los roles del equipo de trabajo, se define líderes y responsables del proyecto. Adicionalmente se orienta metas y objetivos a alcanzar, herramientas a utilizar, el estudio de factibilidad; además de realizar el análisis y gestión de riesgos del proyecto.

1.3.4.2 *Planificación*

En la fase Planificación se inicia con la definición y especificación de requerimientos tanto funcionales y no funcionales, definición de interfaces, actores, modelar los casos de uso, definir la arquitectura apropiada para la solución y realizar estimaciones de costos.

1.3.4.3 *Desarrollo*

En la fase Desarrollo, el equipo realiza la construcción de los componentes de la solución (código); además se desarrolla la infraestructura del sistema.

1.3.4.4 *Estabilización*

Durante la fase Estabilización, el equipo culmina con una revisión del sistema, orienta un plan de pruebas en entornos realistas y prioriza la solución a errores.

1.3.4.5 *Instalación*

Al concluir con la fase Instalación el equipo de trabajo realiza el plan de instalación y soporte, además de entregar los manuales de usuario como guía en la utilización del producto entregado. Al mismo tiempo, espera la aprobación final del cliente.

1.4 **ESTANDAR ISO-9126**

El estandar internacional fue desarrollado en 1991, sirve para evaluar la calidad de los productos de origen software. El estandar ISO-9126 define seis métricas y sub-métricas, las cuales son un conjunto detallado para evaluar cualquier sistema informático.

1.4.1 *Métricas y Sub-métricas*

Las métricas tratan de servir de medio para entender, monitorizar, controlar, predecir y probar el desarrollo de software y de los proyectos de mantenimiento (García, 2010. pp. 7-8).

En la tabla 1-1 se muestran las métricas y sub-métricas del estandar ISO-9126 para evaluar la calidad en métricas de uso del software.

Tabla 1-1: Métricas y Sub-métricas

Métricas	Concepto	Sub-métricas	Descripción
Funcionalidad	Califica el manejo adecuado de las funciones que satisfacen a las necesidades para las cuales fue diseñado el producto de software	Idoneidad	¿Desempeña las tareas para las cuales fue desarrollado?
		Exactitud	¿Tiene el resultado esperado?
		Interoperabilidad	¿Interactúa con otro sistema?
		Seguridad	¿Permite el acceso a personal no autorizado?
Fiabilidad	Califica la capacidad del producto software para mantener su nivel de ejecución en un periodo de tiempo bajo condiciones definidas.	Madurez	¿Verifica y/o Elimina fallas del sistema?
		Recuperabilidad	¿Recupera datos perdidos?
		Tolerancia a fallo:	¿Maneja errores?
Usabilidad	Evalúa el esfuerzo que deberá utilizar el usuario para manejar el sistema.	Aprendizaje	¿Sistema fácil para aprender?
		Comprensión:	¿Sistema fácil de comprender?
		Operatividad	¿Cantidad de esfuerzo para utilizar el sistema?
Eficiencia	Evalúa los aspectos entre el funcionamiento del software y la cantidad de recursos usados.	Comportamiento en el tiempo	¿Responde rápido el sistema?
		Comportamiento de recursos	¿Utiliza los recursos eficientemente?
Mantenibilidad	Mide el esfuerzo necesario para realizar modificaciones al sistema (corrección de errores o incremento funcional)	Estabilidad	¿Mantiene su funcionamiento?
		Facilidad de análisis	¿Diagnostica fácilmente fallas?
		Facilidad de cambio	¿Fácil modificar?
		Facilidad de pruebas	¿Fácil probar?
Portabilidad	Habilidad del software de ser transferido de un ambiente a otro.	Capacidad de instalación	¿Instalar fácilmente?
		Capacidad de reemplazamiento	¿reemplaza otro software similar?
		Adaptabilidad	¿Portable a otros ambientes?
		Co-existencia	¿Funciona con otros sistemas?

Realizado por: Gómez, Diana, 2017.

Estas métricas de calidad se utilizan cuando el producto final es usado en condiciones reales. El estandar ISO-9126 establece una distinción entre validación y verificación.

1.4.2 Usabilidad

Se refiere a la experiencia del usuario a interactuar con un sitio web. Un sitio web con buena usabilidad es aquel que muestra todo el contenido de una forma clara y sencilla de entender por parte del usuario, esto es un aspecto fundamental del software. (Gómez & Alvarado, 2015, p. 24).

La usabilidad se evalúa a través de la norma ISO/IEC 9126, donde la usabilidad es la eficiencia y satisfacción con la que el producto permite alcanzar objetivos específicos a usuarios en un contexto de uso específico. (Gómez & Alvarado, 2015, p. 24).

La usabilidad será el atributo de calidad para evaluar, ya que permite al usuario determinar si la herramienta web podrá ser empleado de manera fácil para su beneficio. Los resultados se verán influenciados por las capacidades del usuario.

1.4.3 Escala de Usabilidad del Sistema (*System Usability Scale, SUS*)

La usabilidad web está definida por la experiencia del usuario frente a una página web. Su utilización en el desarrollo nos permite ver la relación persona-computador, equilibrando la elegancia y necesidad con la que diseñamos la página web. La usabilidad al ser tomada en cuenta en el desarrollo permite reducir costos en asistencia al usuario, mejora la satisfacción del cliente, disminuye los errores del usuario.

Después de análisis del estandar ISO-9126 se utilizará la métrica de usabilidad para evaluar la calidad del sistema de requisiciones a través del test SUS por sus siglas en inglés *System Usability Scale*. SUS es una herramienta estandar en la investigación para medir la usabilidad de un sistema web. Utilizada para resumir las respuestas de los usuarios, debido a su simplicidad y exactitud. Este test consta de un cuestionario de diez preguntas, evaluadas con un sistema de puntuación de 5 puntos que comprende desde “completamente de acuerdo” hasta “completamente desacuerdo”, su tabulación es por un algoritmo rápido. La puntuación de usabilidad es equivalente entre 0 a 100.

En las diez preguntas del test SUS se hace referencia a los siguientes aspectos:

- 1.Creo que me gustaría utilizar este sistema frecuentemente.
- 2.El sistema me resulto innecesariamente complejo.
- 3.Creo que el sistema es bastante fácil de utilizar.
- 4.Creo que necesitaría el soporte de un técnico para poder utilizar este sistema.
- 5.Creo que as diferentes funciones del sistema se encuentran muy bien integradas.
6. Opino que hubo demasiada inconsistencia en el sistema.
7. Imagino que la mayoría de las personas aprendería a utilizar el sistema rápidamente.
8. Me sentí algo incómodo al utilizar este sistema.
9. Me sentí muy seguro al utilizar este sistema.
10. Necesito aprender muchas otras cosas antes de poder utilizar correctamente el sistema.

Al término del test SUS, se realiza una sumatoria con el número generado en cada pregunta del test, formando una medida compuesta de la usabilidad del sistema sometido a estudio. Para obtener la sumatoria, las preguntas impares se tomarán el valor asignado por el usuario menos uno y las preguntas pares será cinco menos el valor asignado por el usuario. Luego de obtener la sumatoria se multiplica por 2.5 para conseguir el porcentaje global SUS.

CAPITULO II

2. MARCO METODOLÓGICO

DESARROLLO DEL SISTEMA DE REQUISICIONES PARA LA EMPRESA HIDROELÉCTRICA ABANICO. S.A. APLICANDO EL ENTORNO DE PROGRAMACIÓN NODE.JS

Este sistema está basado en la metodología MSF (Microsoft Solutions Framework), al ser una metodología tradicional, flexible puede adaptarse a diferentes proyectos sin importar las elecciones de herramientas tecnológicas, por ende, en este capítulo se aborda las fases de desarrollo de MSF, tales como *visión y alcance* donde está dada por la visión clara y conjunta del equipo para lograr satisfacer al cliente y motivar al equipo, a continuación la fase planificación donde se define y especifica requerimientos, posteriormente la fase de desarrollo donde se construye los componentes de la solución, finalmente la fase estabilización donde culmina con una revisión del sistema, y para concluir con la fase Instalación realizando manuales de guía para el usuario.

2.1. Visión y Alcance

2.1.1. *Definición del Problema*

Una requisición de compra (documento) es originada por un empleado en la empresa (planta 1 o planta 2) y enviada a través del transporte hacia la oficina principal ubicada en la ciudad de Macas. Cuando el documento se encuentra en la oficina recibe la autorización por parte del gerente. Una vez autorizada, la secretaria procede al despacho hacia el lugar donde se originó la requisición con el fin de abastecer bienes o servicios requeridos por los trabajadores. Cabe recalcar que el proceso de notificar las requisiciones toma de 1 a 2 días, tampoco existe ninguna revisión de lo

solicitado antes de la autorización por lo cual consume tiempo y recursos ocasionando deficiencia en el control de las requisiciones realizadas.

2.1.2 *Visión del Proyecto*

El principal objetivo es entregar una aplicación web con el propósito de notificar de manera inmediata el pedido de requisición, de esta manera se podría incrementar la comunicación entre el personal administrativo y operativo.

2.1.3 *Perfiles de Usuario*

Administrador

Persona que posea un perfil de: Ingeniero en sistema con conocimientos básicos de redes y conocimiento en lenguaje de programación *Javascript* y *Python*

Funciones:

- Gestionar permisos.
- Gestionar usuarios.
- Gestionar categorías.
- Gestionar productos.
- Gestionar requisiciones.
- Gestionar notificaciones.
- Gestionar reportes.

Administrativo

Perfil de usuario: Empleado con el permiso total en la realización de las requisiciones ej. Secretaria, Gerente.

Tareas:

- Gestión inventario.
- Gestión requisiciones.
- Visualización de reportes
- Impresión de reportes

Operador

Perfil de usuario: Empleado en las centrales con conocimientos básicos en la utilización de equipos de cómputo.

Tareas:

- Gestionar requisiciones.

2.1.4 *Ámbito del Proyecto*

El sistema web que se desarrolla para la empresa Hidroabánico S.A. denominado sistema de requisiciones, el cual ayuda en el control y seguimiento por parte de los administrativos, para el proceso de compra de bienes.

El Sistema de requisiciones brinda un portal de comunicación entre la planta central y la oficina, donde los trabajadores registren las necesidades de compra para su posterior revisión por parte de los administrativos.

2.1.5 *Herramientas a Utilizar*

En la tabla 2-2 se detalla las herramientas a utilizar en el desarrollo del sistema web de requisiciones.

Tabla 2-2: Herramientas

HERRAMIENTA	DESCRIPCIÓN
Node.JS	Framework para desarrollo de aplicaciones web en tiempo real
Java Script & Python	Lenguaje de programación
Django	Framework de desarrollo
PostgreSQL	Base de datos
Windows	Sistema operativo
Ubuntu Server	Servidor
MSF	Metodología de desarrollo de software.
Microsoft Project	Calendarización, cálculo estimado del tiempo
StartUML	Lenguaje gráfico para modelamiento de software
ReportLab	Constructor y diseñador de reportes

Realizado por: Gómez, Diana, 2017.

2.1.6 *Objetivos de Proyecto*

OBJETIVO GENERAL

Desarrollar el sistema de requisiciones para la empresa Hidroeléctrica Abanico. S.A. aplicando el entorno de programación NODE.JS

OBJETIVOS ESPECIFICOS

- Analizar la gestión de requisiciones desde una perspectiva que se ajuste al diseño de una aplicación web.
- Estudiar las herramientas informáticas Python y Javascript del lado del servidor utilizando NODE.JS para el desarrollo de la aplicación web.

- Desarrollar la aplicación web mediante el uso de notificaciones en tiempo real para la Empresa HidroAbanico S.A.
- Evaluar la usabilidad del sistema de requisiciones mediante la aplicación del test SUS a los empleados de la empresa HidroAbanico S.A.

2.1.7 *Análisis y Gestión del riesgo*

La tabla 3-2 muestra los riesgos existentes en el desarrollo de un sistema web para su posterior análisis.

Tabla 3-2: Riesgos

Nº	CLASE	RIESGO	RESULTADO
1	Riesgo Técnico	Herramienta de desarrollo compleja de aprender.	Proyecto con posible retraso
2	Riesgo Técnico	Mala planificación del tiempo para la culminación del proyecto.	Software Incompleto
3	Riesgo Técnico	Requerimientos incorrectos no satisfacen al cliente, replantear requerimientos.	Planificación incorrecta
4	Riesgo del Proyecto	El empleado encargado de la entrega de información ya no pueda continuar.	Proyecto con posible retraso
5	Riesgo del Proyecto	Información no se puede recuperar por daño en computadora.	Proyecto con posible cancelación
6	Riesgos del proyecto	Requerimientos cambian continuamente	Proyecto con posible retraso
7	Riesgo de negocio	Interfaces poco amigables con el usuario, difíciles de manipular.	Tiempo y recursos perdidos.

Realizado por: Gómez, Diana, 2017.

La tabla 4-2 indica la determinación del impacto de los riesgos encontrados en el desarrollo del proyecto.

Tabla 4-2: Impacto Técnico

Valor	Impacto	Efecto en el Desarrollo del Proyecto	Rango
1	Bajo	Ligero (retraso 5 días)	<5%
2	Moderado	Moderado (retraso 10 días)	5-10%
3	Alto	Severo (retraso 15 días)	11-20%
4	Muy Alto	No culminación del proyecto (retraso 20 días)	>20%

Realizado por: Gómez, Diana, 2017.

La tabla 5-2 muestra el rango de probabilidad de ocurrencia de los riesgos.

Tabla 5-2: Probabilidad

Valor	Probabilidad	Rango
1	Baja	1% -40 %
2	Media	41% - 70%
3	Alta	71% - 100%

Realizado por: Gómez, Diana, 2017.

Las tablas 6-2 y 7-2 indican la exposición al riesgo y el código de colores para determinar la categoría de los riesgos.

Tabla 6-2: Exposición al riesgo

Exposición			
Exposición	Baja	Media	Alta
Valor	1-2	3-4	>6

Realizado por: Gómez, Diana, 2017.

Tabla 7-2: Código de colores

Impacto				
4	Crítico	12	8	4
3	Alto	9	6	3
2	Moderado	6	4	2
1	Baja	3	2	1
		Baja	Media	Alta
		Probabilidad		

Realizado por: Gómez, Diana, 2017.

2.1.7.1 Desarrollo del análisis del Riesgo

En la tabla 8-2 se indica como el investigador define parámetros referentes al impacto técnico y la probabilidad de ocurrencia de los riesgos en las tablas 4-2 y 5-2. Además, define los parámetros de las tablas 6-2 y 7-2 basadas en el cálculo de la prioridad y exposición al riesgo.

Tabla 8-2: Análisis del riesgo

ID RIESGO	R1	R2	R3	R4	R5	R6	R7
Valor	3	3	2	2	4	1	1
Impacto	Moderado	Alto	Moderado	Moderado	Muy Alto	Bajo	Bajo
%	60	30	30	70	60	30	25
Valor	2	1	1	3	2	1	1
Probabilidad	Media	Baja	Media	Alta	Media	Baja	Baja
Valor	6	3	3	6	8	1	1
Exposición	Alta	Media	Media	Alta	Alta	Baja	Baja

Realizado por: Gómez, Diana, 2017.

2.1.7.2 Línea de Corte

En la tabla 9-2 se divide los riesgos más expuestos por código de colores basándose en la exposición para su posterior atención y planificación.

Tabla 9-2: Línea de corte

IDENTIFICACIÓN	R1	R2	R3	R4	R5	R6	R7
EXPOSICIÓN	6	8	6	3	3	1	1

Realizado por: Gómez, Diana, 2017.

Desde las tablas 10-2 hasta la tabla 12-2 se detalle cada uno de los riesgos mostrados en la línea de corte con su debida planificación para gestionar el riesgo.

Tabla 10-2: Riesgo 1

HOJA DE GESTIÓN DE RIESGO	
ID RIESGO	R1
FECHA	
PROBABILIDAD Media (2)	DESCRIPCIÓN Herramienta de desarrollo compleja de aprender.
IMPACTO Moderado (3)	REFINAMIENTO Causa: La desarrolladora del proyecto no domina el lenguaje de programación Node.JS – Java Script. Consecuencias: <ul style="list-style-type: none"> • Proyecto con posible retraso • Poca información en español REDUCCIÓN: La investigadora consulta a una persona que domine la herramienta de desarrollo y le pueda ayudar con pautas sobre utilización de la herramienta. SUPERVISIÓN Constatar que se entregue a tiempo las tareas. Planificar la entrega de tareas retrasadas. GESTIÓN Buscar información sobre la manipulación del lenguaje de programación ESTADO ACTUAL: <div> <input checked="" type="checkbox"/> Fase reducción Fase Supervisión Gestión del Riesgo </div> RESPONSABLE: Diana E. Gómez G.
EXPOSICIÓN Alta (6)	
PRIORIDAD 2	

Realizado por: Gómez, Diana, 2017.

Tabla 11-2: Riesgo 3

HOJA DE GESTIÓN DE RIESGO	
ID RIESGO	R2
FECHA	
PROBABILIDAD Media (2)	DESCRIPCIÓN Mala planificación del tiempo para la culminación del proyecto.
IMPACTO Muy Alta (4)	REFINAMIENTO Causa: Planificar sin tomar en cuenta la complejidad del proyecto. Consecuencias: <ul style="list-style-type: none"> • Proyecto con posible retraso • Proyecto Incompleto
EXPOSICIÓN Alta (8)	
PRIORIDAD 1	
	REDUCCIÓN: Aumento de esfuerzo por parte de la desarrolladora para terminar el proyecto.
	SUPERVISIÓN Tareas y entregables deben cumplir planificación de fechas.
	GESTIÓN Cumplir las tareas establecidas para no retrasar el proyecto.
	ESTADO ACTUAL: <div> <input checked="" type="checkbox"/> Fase Reducción </div> <div> <input type="checkbox"/> Fase Supervisión </div> <div> <input type="checkbox"/> Gestión del Riesgo </div>
	RESPONSABLE: Diana E. Gómez G.

Realizado por: Gómez, Diana, 2017.

Tabla 12-2: Riesgo 5

HOJA DE GESTIÓN DE RIESGO	
ID RIESGO	R3
FECHA	
PROBABILIDAD Media (2)	DESCRIPCIÓN Requerimientos incorrectos no satisfacen al cliente, replantear requerimientos.
IMPACTO Muy Alta (4)	REFINAMIENTO Causa: Iniciar el proyecto con requerimientos incorrectos, dudas presentes en la desarrolladora. Consecuencias: <ul style="list-style-type: none"> • Proyecto con posible retraso REDUCCIÓN: Entrevista con el cliente para redefinir los requerimientos. SUPERVISIÓN Guía del tutor hacia la investigadora. GESTIÓN Despejar dudas sobre los requerimientos con preguntas constantes hacia el cliente. ESTADO ACTUAL: <div> <input checked="" type="checkbox"/> Fase Reducción <div> <input type="checkbox"/> Fase Supervisión <div> <input type="checkbox"/> Gestión del Riesgo </div> </div> </div> RESPONSABLE: Diana E. Gómez G.
EXPOSICIÓN Alta (8)	
PRIORIDAD 1	

Realizado por: Gómez, Diana, 2017.

2.1.8 *Planificación Inicial*

DEFINICIÓN DEL PLAN BORRADOR

Sistema de Requisiciones Hidroabanico S.A.

Sector Solicitante: Empresa Hidroeléctrica Abanico S.A.

Equipo de trabajo: Los roles a cumplir por parte de la investigadora se muestra en la tabla 13-2

Tabla 13-2: Equipo de Trabajo

INTEGRANTE	CARGOS
Diana E. Gómez G.	Gestora del Proyecto
	Desarrolladora

Realizado por: Gómez, Diana, 2017.

Aspectos Generales de la Empresa

Nombre de la empresa: Empresa Hidroeléctrica Abanico S.A.

Dirección: Provincia Morona Santiago, Cantón Morona, Parroquia Macas.

Teléfono: (593-2)395-7521

(593-2) 394-5783

Fax: (593-2) 394-5765

Código Postal: 140150

Marco Legal

“Hidroabanico S.A. cumplió con todos los trámites legales para obtener los permisos para la construcción, instalación y operación de la Central Hidroeléctrica Abanico de acuerdo con lo que establece la Ley de Régimen del Sector Eléctrico.

Previo al inicio de la construcción, en septiembre del 2004, Hidroabanico S.A. obtuvo y firmó el contrato de permiso, la autorización del Conelec, el contrato de garantía de inversiones y el permiso de aguas”.

Visión

“Generar y comercializar energía limpia con altas normas de calidad y eficiencia, con responsabilidad social y ambiental, con un equipo humano calificado y comprometido con el futuro del país, aportando así de manera efectiva al desarrollo local y nacional”.

Misión

“Ser la empresa privada más importante del país que aporta al crecimiento del Ecuador con la generación de energía limpia de manera eficiente y creando valor para un desarrollo responsable del país. Lograr las mejores prácticas del sector en beneficio de sus clientes, colaboradores y accionistas”.

Valores

“Compromiso con el Ecuador

Ética

Integridad y Confianza

Respeto por las personas

Honestidad

Iniciativa y trabajo en equipo

Conciencia ambiental

Servicio a la comunidad

Calidad y eficiencia

Responsabilidad social

Profesionalismo”

ORGÁNICO FUNCIONAL

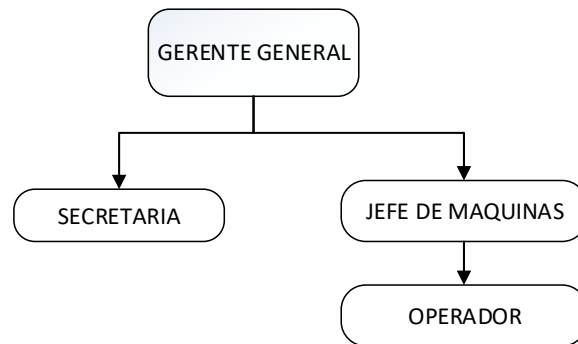


Figura 11-2: Orgánico Funcional – Cantón Morona
Realizado por: Gómez, Diana, 2017.

GENERALIDADES

Fuente de Recopilación de la Información

Para el caso de esta investigación es considerado el personal administrativo (ej. Secretaria) y operativo (ej. Operador), quienes provee la información necesaria para construir la base de datos y definir el correcto funcionamiento del sistema. El personal administrativo tiene conocimiento sobre el proceso de requisiciones de compra, es quien utiliza el sistema en la oficina central. El personal operativo tiene conocimiento de la creación de la requisición, es quien utiliza el sistema desde la planta.

TÉCNICA A UTILIZAR

Técnica de Recopilación de la Información

Entrevistas

- Se debe planificar entrevista con el personal administrativo fuente de información (Secretaria).
- Se debe realizar diálogos con los operadores para establecer requerimientos que desean ver plasmados en el sistema.

Al personal administrativo para recopilar información. La entrevista realizada se encuentra en el **anexo A** de este documento.

Lluvia de Ideas (Presentación de la propuesta)

Se debe planificar reuniones con el personal de la empresa en donde surge propuestas e ideas, las cuales se pondrán en discusión y análisis y de esta manera establecer de la mejor forma todas las necesidades y requerimientos que el cliente quiere ver en el sistema a desarrollar.

Técnica para representar actividades del sistema

Diagrama de actividades

Se determina un diagrama de actividades con el cual se podrá cumplir con una serie de contenidos tomando en cuenta una secuencia determinada de procesos y un tiempo establecido para cada uno de ellos.

SISTEMAS INFORMÁTICOS

De manera resumida se muestran en las tablas 14-2 hasta la 19-2 los procesos principales del sistema de requisiciones.

Tabla 14-2: Creación de cuenta

ACTIVIDAD	ACTOR
Ingreso al sistema	Secretaria (Oficina)
Creación de la cuenta del personal administrativo y operadores que utilizaran el sistema	Secretaria o Desarrollador
Valida información	Sistema de Requisiciones

Realizado por: Gómez, Diana, 2017.

Tabla 15-2: Relleno de la requisición de compra

ACTIVIDAD	ACTOR
Ingreso al sistema	Operador (Planta)
Ingreso con los permisos respectivos, Ingreso de datos respectivos de la requisición en el sistema.	Empleado
Valida información	Sistema de Requisiciones

Realizado por: Gómez, Diana, 2017.

Tabla 16-2: Autorización de las requisición de compra

ACTIVIDAD	ACTOR
Ingreso al sistema	Secretaria (Oficina)
Ingreso con los permisos respectivos, consulta la notificación de la requisición.	Secretaria
Valida información	Sistema de Requisiciones

Realizado por: Gómez, Diana, 2017.

Tabla 17-2: Consulta de reportes

ACTIVIDAD	ACTOR
Ingreso al sistema	Secretaria (Oficina)
Ingreso con los permisos respectivos, consulta los reportes requeridos.	Secretaria
Valida información	Sistema de Requisiciones

Realizado por: Gómez, Diana, 2017.

Tabla 18-2: Generación de reportes

ACTIVIDAD	ACTOR
Ingreso al sistema	Secretaria (Oficina) Empleado (Planta)
Ingreso con los permisos respectivos. Generación de reportes requeridos.	Secretaria
Valida información	Sistema de Requisiciones

Realizado por: Gómez, Diana, 2017.

Tabla 19-2: Registro de Nuevos Usuarios

ACTIVIDAD	ACTOR
Ingreso al sistema	Secretaria (Oficina)
Creación de la cuenta y contraseña para el respectivo registro en el sistema.	Secretaria
Valida información	Sistema de Requisiciones

Realizado por: Gómez, Diana, 2017.

DIAGRAMA DE ACTIVIDADES DEL SISTEMA

En la figura 12-2 se indica la nomenclatura de los siguientes diagramas basados en *flujogramas*.

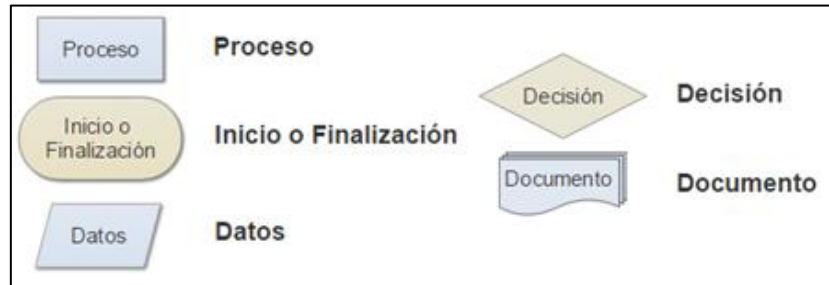


Figura 12-2: Nomenclatura de flujogramas

Realizado por: Gómez, Diana, 2017.

Proceso: Procesan información recibida.

Inicio o finalización: Muestra el principio y la terminación de un diagrama.

Datos: Ingreso de información al sistema.

Decisión: Condicionante para la toma de decisiones.

Documento: Imprime algún documento creado por el sistema.

En las figuras 13-2 hasta la 18-2, muestran los diagramas de flujo de las actividades del sistema.

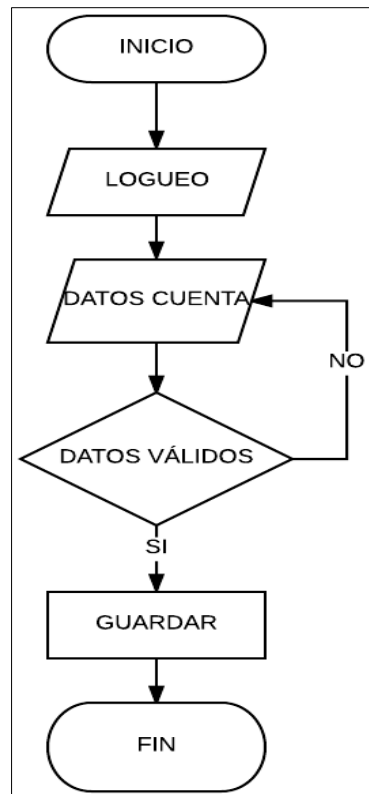


Figura 13-2: Creación de la cuenta
 Realizado por: Gómez, Diana, 2017.

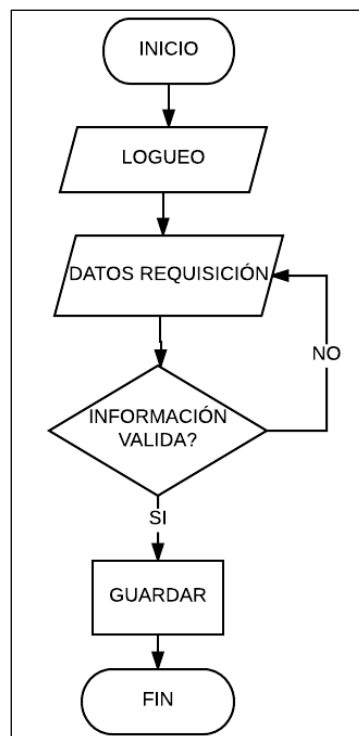


Figura 14-2: Relleno de la requisición
 Realizado por: Gómez, Diana, 2017.

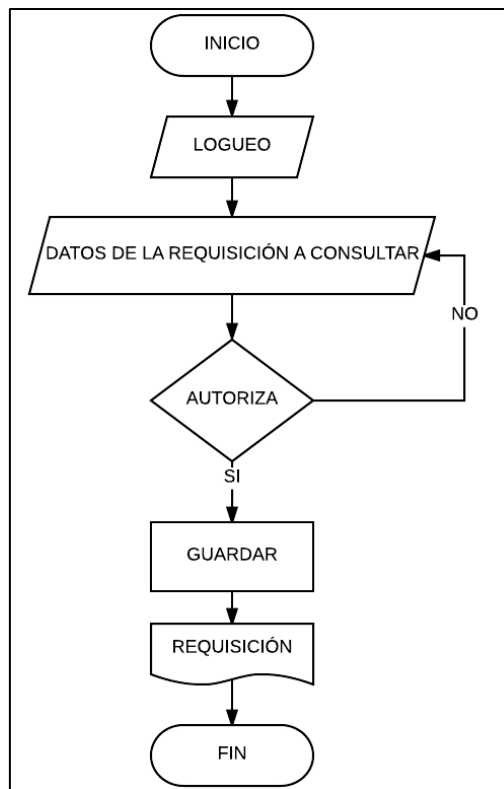


Figura 15-2: Autorización de la requisición
 Realizado por: Gómez, Diana, 2017.

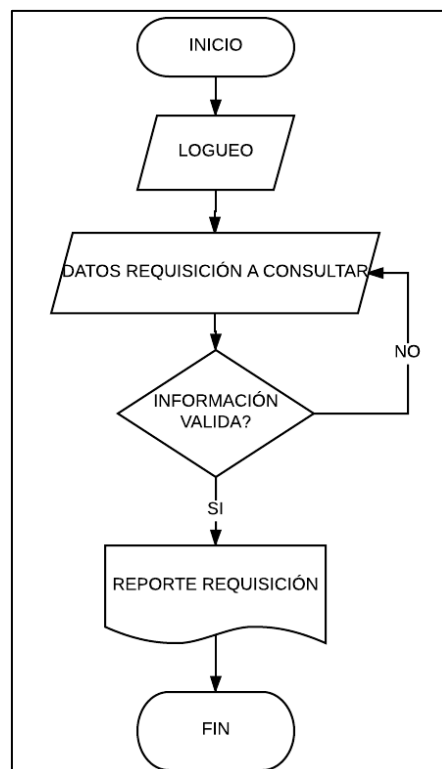


Figura 16-2: Consulta de reportes
 Realizado por: Gómez, Diana, 2017.

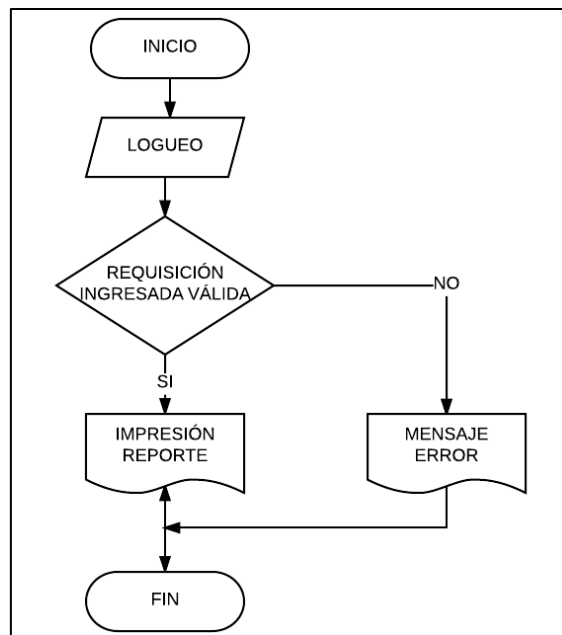


Figura 17-2: Generación de reportes

Realizado por: Gómez, Diana, 2017.

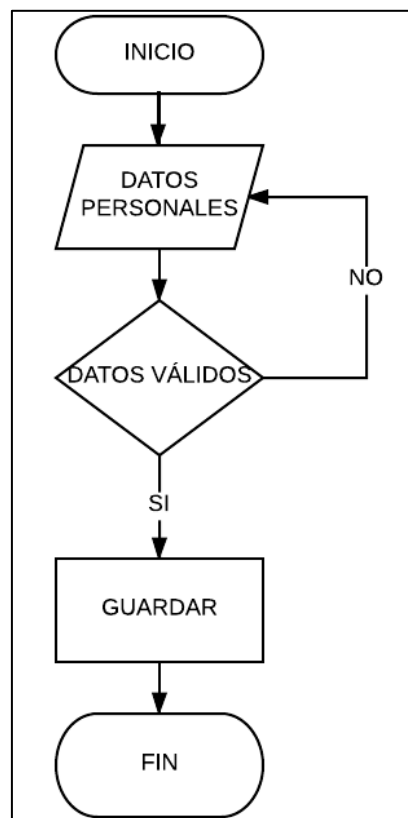


Figura 18-2: Registro de Nuevo Usuario

Realizado por: Gómez, Diana, 2017.

DIAGRAMA GANTT

Según la metodología MSF, se enlista las fases cronológicamente para el desarrollo del sistema de requisiciones, tal como se muestra en el **anexo B** de este documento.

2.1.9 *Estudio de Factibilidad*

2.1.9.1 *Factibilidad Técnica*

Hardware Requerido

1 LAPTOP

Procesador AMD E2-7110 APUwith AMD Radeon R2 Graphics 1.80 GHz

Memoria RAM: 4GB

Sistema Operativo: Windows 10 de 64 bits

Disco Duro: 500GB

1 SERVIDOR DE APLICACIONES

Servidor Web Ubuntu-512mb-nyc3-01

Memoria RAM: 512 MB

Disco Duro: 20 GB

Software Mínimo Requerido

Navegador Web

Herramientas Utilizadas

SublimeText

Django

Node.JS

PostgreSQL 9.2

ReportLab

2.1.9.2 *Factibilidad Operativa*

1. Se debe capacitar al personal administrativo(Secretaria) y personal operativo(Operador) para usar el sistema de requisiciones de una manera correcta.
2. Se requiere de personal experto en el desarrollo de software para la administración y mantenimiento del sistema de requisiciones.

2.1.9.3 *Factibilidad Legal*

El sistema de requisiciones no se involucra en la toma de decisiones por parte de los directivos, proceso de compra de artículos, selección de proveedores; sirve de ayuda en el proceso de control y despacho de las requisiciones.

2.1.9.4 *Factibilidad Económica*

La tabla 20-2 indica detalladamente el presupuesto del proyecto mostrando los recursos necesarios para la realización del sistema.

Tabla 20-2: Presupuesto Proyecto

	Descripción	Precio U.	Precio Total
Recurso Humano	Desarrollador por 1 año	500 c/mes	6000,00
Hardware	Servidor (1 año)	50,00 c/mes	600,00
	Laptop	1250,00	1250,00
Otros Gastos	Transporte (1 año)	15 c/mes	180,00
	Internet (1 año)	25 c/mes	300,00
	Telefonía (1 año)	10 c/mes	120,00
	Oficina (Insumos)	250	250,00
	Electricidad	20 c/mes	240,00
Costo Total del Proyecto (Sin IVA)			8940,00

Realizado por: Gómez, Diana, 2017.

2.2 Planificación

2.2.1 Definición y Especificación de Requerimientos

Objetivos del SRS

Ofrece tanto al usuario y desarrollador un punto de inicio; brinda una visión de lo deseado durante el desarrollo del sistema. Se considera las necesidades, limitaciones, requerimientos de usuario, hardware y software.

Ámbito

La empresa ha aceptado el proyecto de titulación con tema “Desarrollo del sistema de requisiciones para la empresa hidroeléctrica abanico. s.a. aplicando el entorno de programación node.js”, el mismo que está enfocado en servir de apoyo para el proceso de notificaciones en la gestión de requisiciones.

El sistema está orientado al proceso de notificaciones entre el personal administrativo y operativo en la toma de decisiones en el proceso de requisiciones que se generen.

Cabe recalcar que el personal administrativo es la secretaria, quien podrá recibir las notificaciones de las requisiciones hecha por el personal operativo desde la planta para su posterior autorización y despacho.

Visión General del SRS

Este documento permite ver la funcionalidad del sistema, conoce las actividades y procesos para la culminación de su desarrollo.

Descripción General del SRS

Se ha propuesto un sistema web como un software de apoyo en la notificación de pedido de requisiciones, este sistema está enfocado en las notificaciones como un evento en tiempo real entre el operador y secretaria, para acelerar el proceso de pedido de requisición.

Razones que motivan la realización del Proyecto

- Mejorar el manejo de requisiciones.
- Mejorar el proceso de notificaciones de pedido de requisición.
- Toma decisiones oportunas por parte de los directivos a través de la generación de reportes.

Perspectiva del producto

Este producto, es un sistema prototipo de interés para la utilización en instituciones públicas y privadas que manejen pedidos de requisición. El sistema sirve de ayuda con la administración de requisiciones mediante el envío de notificaciones, utilizado un navegador web.

Funciones del Producto

Autenticación

- Creación – Modificación – Eliminación de cuentas a usuarios del sistema
- Asignación de Perfiles de Usuario (Administrativo - Operador)

Inventario

- Gestión de categorías
- Ingreso de Información Categorías (categoría y subcategoría)
- Gestión de Productos
- Ingreso de Información Productos (categoría, nombre producto, stock, nota)

Requisiciones

- Gestión de Lugar
- Ingreso de Información Lugar (nombre)
- Gestión de Pedidos
- Ingreso de Información Pedidos (cantidad, producto, lugar, nota general)

Notificaciones

- Generar notificaciones en tiempo real entre el generador de la requisición y quien autoriza la misma.

Reportes

- Visualización e impresión (estado, rango de fechas, empleado)

La palabra gestión define las siguientes acciones: creación, modificación y eliminación.

Características del Usuario

Los usuarios se categorizan en:

- Administradores
- Personal Administrativo (Secretaria - Gerente)
- Personal Operativo (Operador)

Limitaciones Generales

- El sistema no soporta administración de proveedores, no calcula costos.

Supuestos y Dependencias

- Añadir nuevas, a las funcionalidades definidas
- El llenado de la base de datos debe ser con información correcta.

2.2.1.1 Interfaces

Interfaz de Usuarios

Los usuarios (Administrador, Administrativos, Operadores) poseen el acceso a las interfaces dependiendo de los permisos de ingreso. Gestión de usuarios, gestión de requisiciones, consulta de notificaciones, generación de reportes; estas interfaces son visualizadas en la web. Se maneja mensajes para verificar la información para su posterior envío a la base de datos.

Interfaz de Hardware

En la tabla 21-2 se indica las características del servidor de aplicaciones para el despliegue del sistema.

Tabla 21-2: Requisitos Servidor

Características	Sistema Operativo	Memoria RAM	Disco Duro
Detalle	Ubuntu	512 MB	20 GB

Realizado por: Gómez, Diana, 2017.

En la tabla 22-2 se indica las características de la computadora cliente para el uso del sistema

Tabla 22-2: Requisitos Cliente

<div>Detalle</div> <div>Características</div>	Procesador	RAM	Disco Duro
Computador	1.8 GHZ	1 GB	500 GB
Acceso Internet	512 Kbps		

Realizado por: Gómez, Diana, 2017.

Impresora

El personal administrativo necesita una impresora para poder archivar de forma física la información de las requisiciones

2.2.1.2 Interfaces de Software

En la Tabla 23-2 indica la instalación de la plataforma de software necesaria.

Tabla 23-2: Interfaces de Software

Tipo	Software
Sistema Operativo Servidor	Ubuntu
DBMS	PostgreSQL
Framework JavaScript	NodeJS
Framework Python	Django
Plataforma de aplicaciones	Mozilla Firefox Google Chrome Internet Explorer.

Realizado por: Gómez, Diana, 2017.

Navegador

Al ser un proyecto web, el requisito principal para el uso del sistema de requisiciones es cualquier navegador web.

2.2.1.3 *Interfaces de Comunicación*

Para el funcionamiento del sistema se recomienda el uso de nginx, que es un servidor ligero de alto rendimiento, de software libre y código abierto; el cual permite el despliegue de la aplicación web.

2.2.1.4 *Requerimientos Funcionales del Sistema*

Los requerimientos funcionales del sistema de requisiciones se los ha organizado en cinco módulos, donde se detalla entradas, procesos y salidas.

- 1.** Gestión de Usuarios
- 2.** Gestión de Inventario
- 3.** Gestión de Requisiciones
- 4.** Gestión de Notificaciones
- 5.** Reportes

Requerimiento 1: Gestión de Usuarios

Descripción: El módulo de gestión de usuarios es administrado bajo el perfil de superusuario o administrativo donde se creará usuario, contraseña y se asigna su perfil de usuario.

Entradas

- Datos usuario: Cedula y contraseña
- Datos información personal: nombre, apellido, correo electrónico.

Procesos

- Sistema permite crear, modificar o eliminar usuarios.
- Sistema permite asignar perfil a los usuarios.
- Sistema permite ingresar los datos básicos del usuario.
- Sistema valida la información.

Salida

- Confirmación: Éxito o fallo

Requerimiento 2: Gestión de Inventario

Descripción: El módulo de inventario es administrado bajo el perfil de superusuario o administrativo donde se ingresa, modifica o elimina productos o categorías.

Entradas

- Datos categoría: Nombre, Categoría
- Datos producto: Nombre, Categoría.
- Datos Ingresos: Stock, Descripción.

Procesos

- Sistema permite crear, modificar o eliminar productos.
- Sistema permite crear, modificar o eliminar categorías.
- Sistema permite crear, modificar o eliminar ingresos.
- Sistema valida la información.

Salidas

- Confirmación: Éxito o fallo

Requerimiento 3: Gestión de Requisiciones

Descripción: El módulo de requisiciones es administrado bajo el perfil operativo o administrativo donde se ingresa, modifica o elimina pedidos de requisición.

Entradas

- Datos de pedido: Producto, cantidad, descripción, Lugar

Procesos

- Sistema permite elegir los productos del inventario.
- Sistema permite ingresar cantidad de producto solicitado.
- Sistema permite elegir el lugar de entrega.
- Sistema permite ingresar una descripción de la requisición.
- Sistema valida la información.

Salidas

- Confirmación: Éxito o fallo

Requerimiento 4: Gestión de Notificaciones

Descripción: El sistema debe generar notificaciones entre el personal operativo que radica en la planta y el personal administrativo que se encuentra en la oficina central.

Entradas

- Generar pedido

Procesos

- Sistema genera la notificación de un pedido de requisición.
- Sistema envía la notificación de aviso a la secretaria.
- Perfil Administrativo procesa el pedido.
- Sistema envía la notificación del procesamiento del pedido al operador.
- Perfil Administrativo envía o rechaza lo solicitado.
- Sistema envía la notificación de entrega o rechazo de requisición.
- Perfil Operativo envía la notificación de recibido.

Salidas

- Muestra estado de la requisición

Requerimiento 5: Reportes

Descripción: El sistema permite la visualización de reportes.

Entradas

- Datos Logueo: Usuario y Contraseña.
- Datos Generación de reportes: Selección de fecha, estado o empleado

Procesos

- Perfil Administrativo selecciona el reporte requerido.

Salidas

- Confirmación: Visualización de reporte.

2.2.1.5 Requerimientos No Funcionales

- Facilidad de uso
- Adaptabilidad
- Disponibilidad
- Escalabilidad

2.2.1.6 Actores

Existen 3 tipos de perfiles de usuarios que utilizaran el sistema:

- Administrador
- Administrativos
- Operativos

2.2.1.7 Casos de Uso

Se muestra el caso de uso del Requerimiento 4 que se refiere a la gestión de notificaciones, los casos de uso de todos los requerimientos se encuentran en el **anexo C** de este documento.

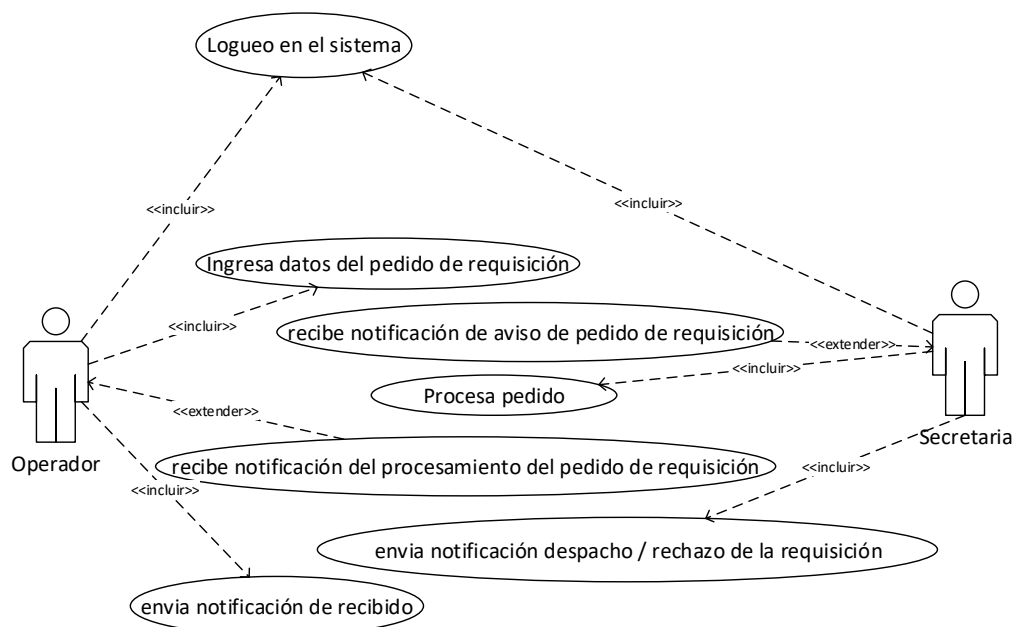


Figura 19-2: Requerimiento # 4 - Caso de Uso - Gestión de Notificaciones

Realizado por: Gómez, Diana, 2017.

Tabla 24-2: Requerimiento #4 - Caso de uso - Gestión de Notificaciones

Número: Requerimiento 4	Usuario: Operador - Secretaria
Nombre historia: Gestión de Notificaciones	
Responsable: Diana E. Gómez G.	
Descripción: El sistema debe generar notificaciones entre el personal operativo que radica en la planta y el personal administrativo que se encuentra en la oficina central	
ACCIONES DE ACTORES	RESPUESTA DEL SISTEMA
Ingresa al sistema	Valida perfil de usuario.
Operador ingresa datos del pedido de requisición	Valida campos, guarda información en base de datos.
Secretaria recibe notificación de aviso / pedido de requisición	Envía notificación a la secretaria.
Secretaria procesa el pedido	Valida entradas y ejecuta la acción.
Operador recibe notificación del procesamiento.	Envía notificación al operador.
Secretaria notifica despacho o rechazo de la requisición.	Envía notificación al operador.
Operador notifica la recepción de la requisición.	Envía notificación a la secretaria.
Observaciones: Usuario ingresa información incorrecta el sistema informa y no ejecuta la acción	

Realizado por: Gómez, Diana, 2017.

2.2.1.8 Arquitectura apropiada para la solución

La arquitectura interna para el sistema de requisiciones se indica en la figura 20-2.

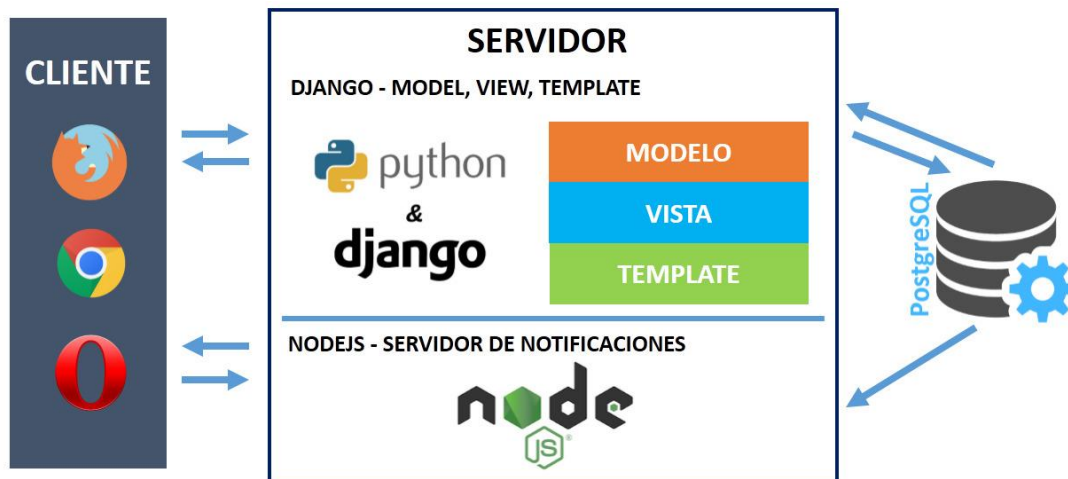


Figura 20-2: Arquitectura interna

Realizado por: Gómez, Diana, 2017.

En las figuras 21-2 y 22-2 se muestra una vista más funcional del sistema:

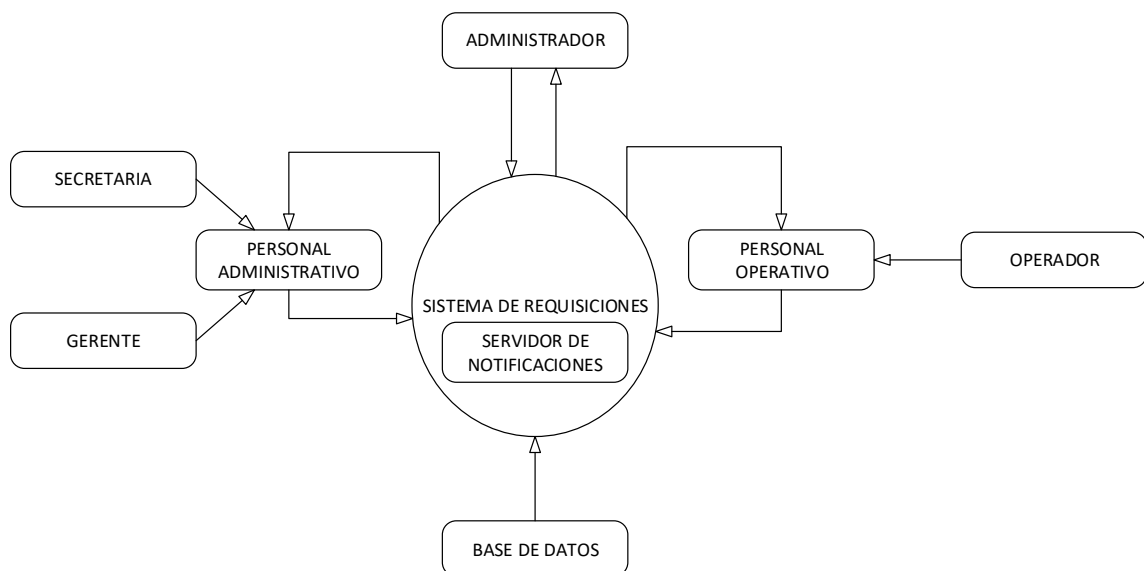


Figura 21-2: Arquitectura externa

Realizado por: Gómez, Diana, 2017.

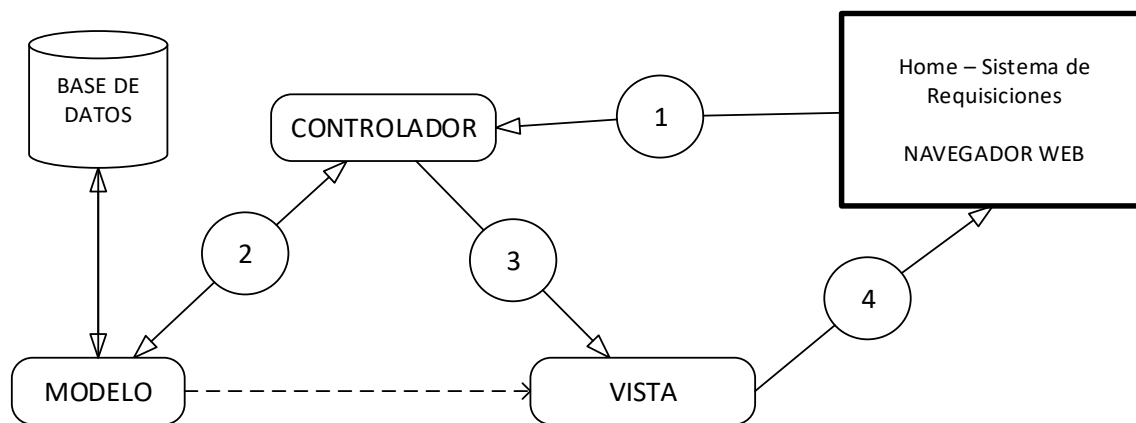


Figura 22-2: Arquitectura externa
Realizado por: Gómez, Diana, 2017.

En la figura 23-2, se muestra la arquitectura detallada.

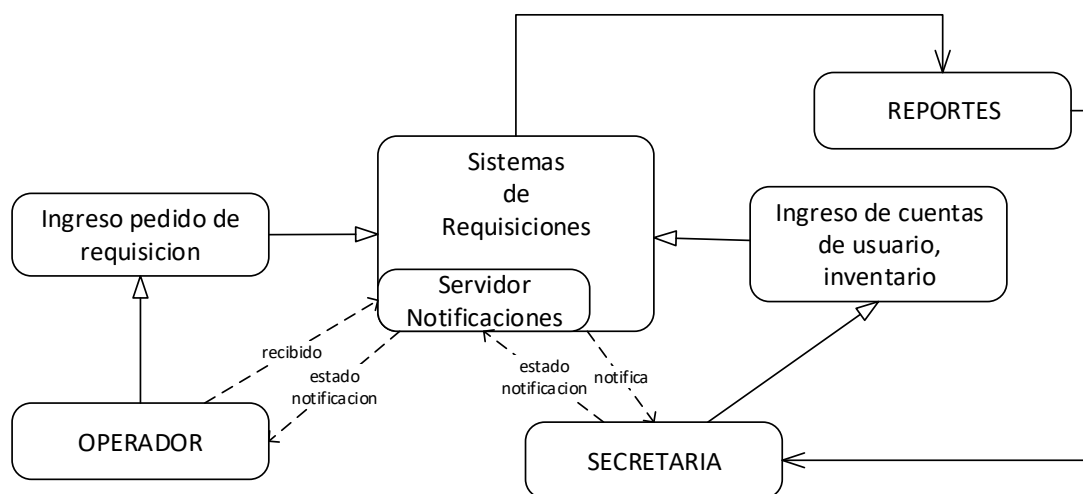


Figura 23-2: Arquitectura detallada
Realizado por: Gómez, Diana, 2017.

2.2.2 Diseño Lógico

2.2.2.1 Tecnología a utilizar en el proyecto

Las herramientas utilizadas para el desarrollo de la aplicación se muestran en detalle en la tabla 29-2.

Tabla 25-2: Herramientas

HERRAMIENTA	CARACTERÍSTICA
Java Script, Python	Lenguaje de Programación
Django 1.11.7	Framework
NodeJS	Marco de trabajo para crear el servidor de notificaciones
PostgreSQL 9.6	Sistema de gestión de Base de datos.
pgAdmin 3	Diseñador y Modelador de base de datos
nginx	Servidor web
ReportLab	Diseñador de informes visual
Microsoft Word 2016	Creación y Edición de documentos
Microsoft Project 2016	Administrador de Proyectos
Microsoft Visio 2016	Diseñador de diagramas UML

Realizado por: Gómez, Diana, 2017.

2.2.2.2 Diagramas de Secuencia

El diagrama de secuencia detallado en la figura 24-2 muestra el flujo de información del sistema de requisiciones para el requerimiento 1. Que trata sobre la gestión de usuarios.

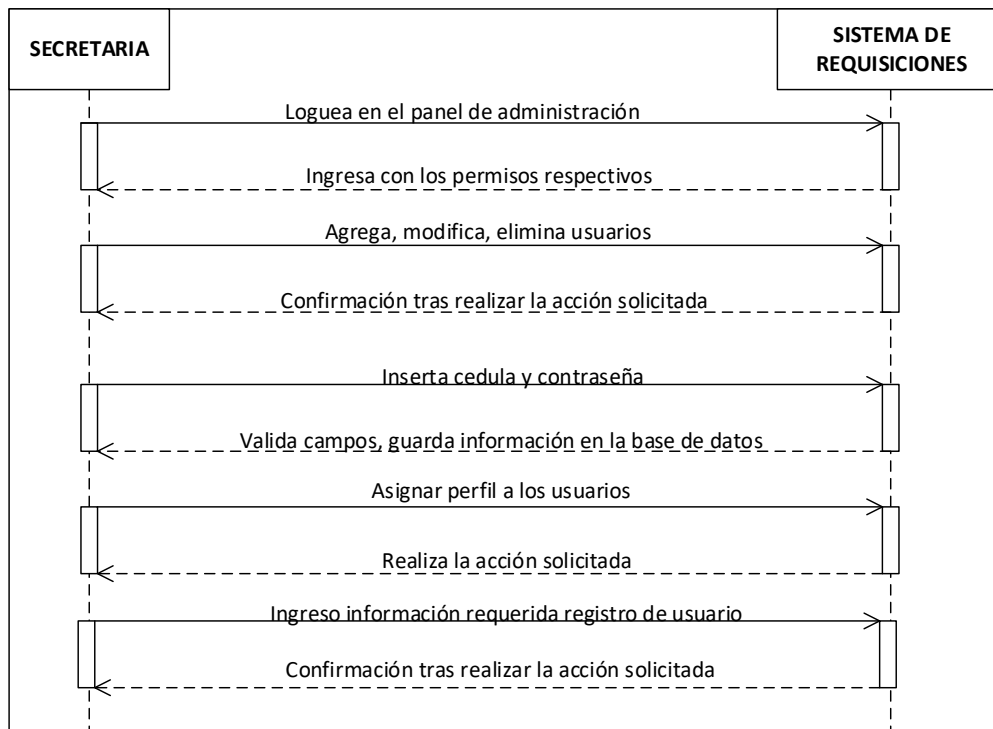


Figura 24-2: Gestión de usuarios - Diagrama de Secuencia Requerimiento #1

Realizado por: Gómez, Diana, 2017.

El diagrama de secuencia detallado en la figura 25-2 muestra el flujo de información del sistema de requisiciones para el requerimiento 2. Que trata sobre la gestión de inventario donde se detalla el ingreso, modificación, eliminación de los productos por parte de la secretaria.

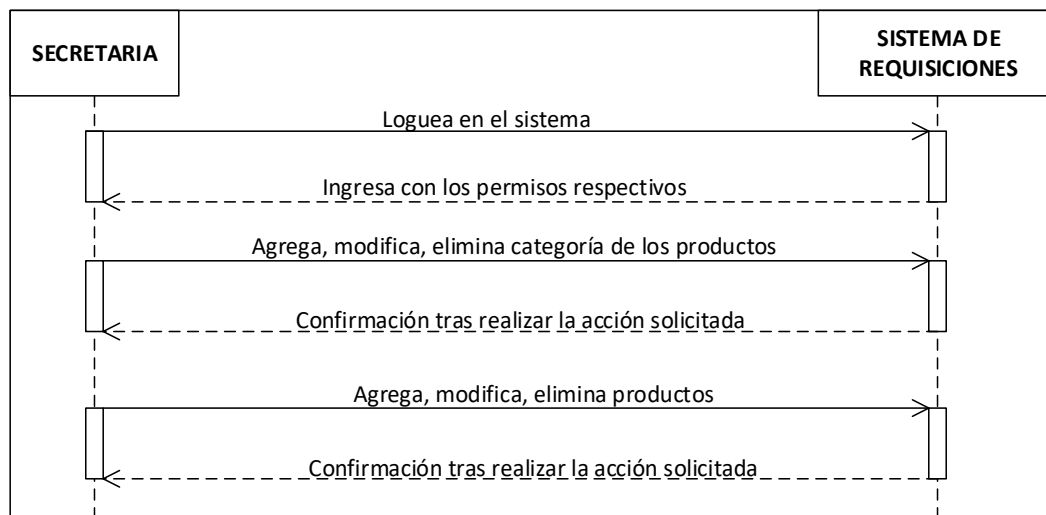


Figura 25-2: Gestión de inventario - Diagrama de secuencia Requerimiento #2

Realizado por: Gómez, Diana, 2017.

El diagrama de secuencia detallado en la figura 26-2 muestra el flujo de información del sistema de requisiciones para el requerimiento 3. Que trata sobre la gestión de requisiciones donde se detalla la gestión de los pedidos realizados por el operador.

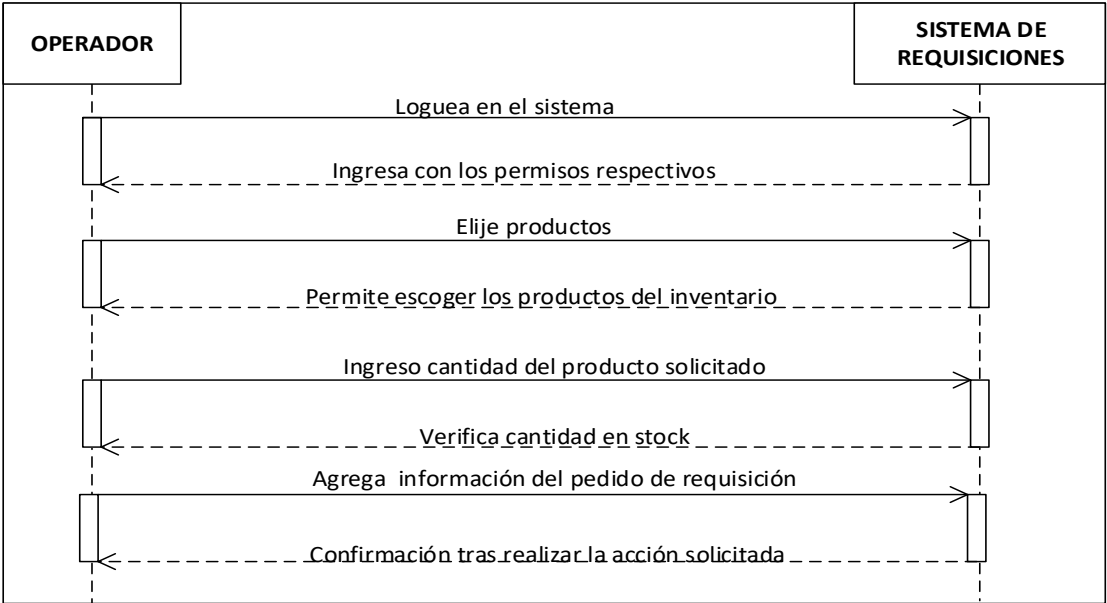


Figura 26-2: Gestión de requisiciones - Diagrama de secuencia Requerimiento #3

Realizado por: Gómez, Diana, 2017.

El diagrama de secuencia detallado en la figura 27-2 muestra el flujo de información del sistema de requisiciones para el requerimiento 4. Que trata sobre la gestión de notificaciones donde se detalla el ingreso de los productos al sistema por parte de la secretaria.



Figura 27-2: Gestión de notificaciones - Diagrama de secuencia Requerimiento #4

Realizado por: Gómez, Diana, 2017.

El diagrama de secuencia detallado en la figura 28-2 muestra el flujo de información del sistema de requisiciones para el requerimiento 5. Que trata sobre los reportes donde se detalla la generación de reportes por parte de la secretaria.

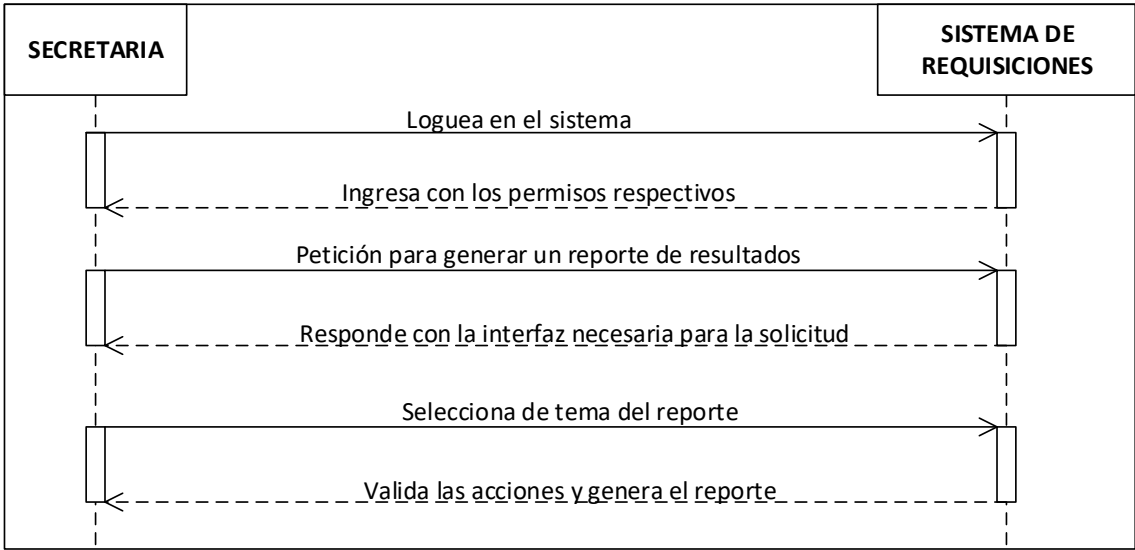


Figura 28-2: Reportes - Diagrama de secuencia Requerimiento #5

Realizado por: Gómez, Diana, 2017.

2.2.2.3 *Diagramas de Clases*

En la figura 29-2 se observa las entidades del sistema de requisiciones, con sus respectivos atributos y operaciones.

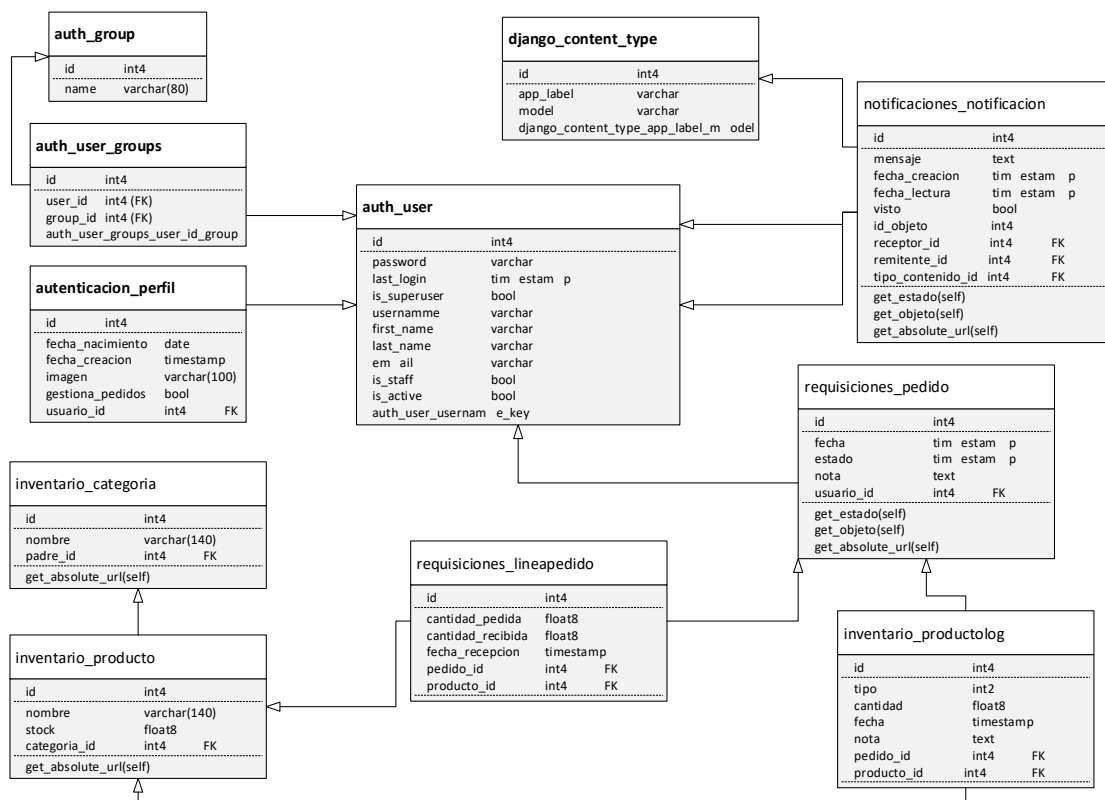


Figura 29-2: Diagrama de clases

Realizado por: Gómez, Diana, 2017.

2.2.2.3 Diagrama de Objetos

En la figura 30-2 se observa la interacción de los objetos del sistema de requisiciones.

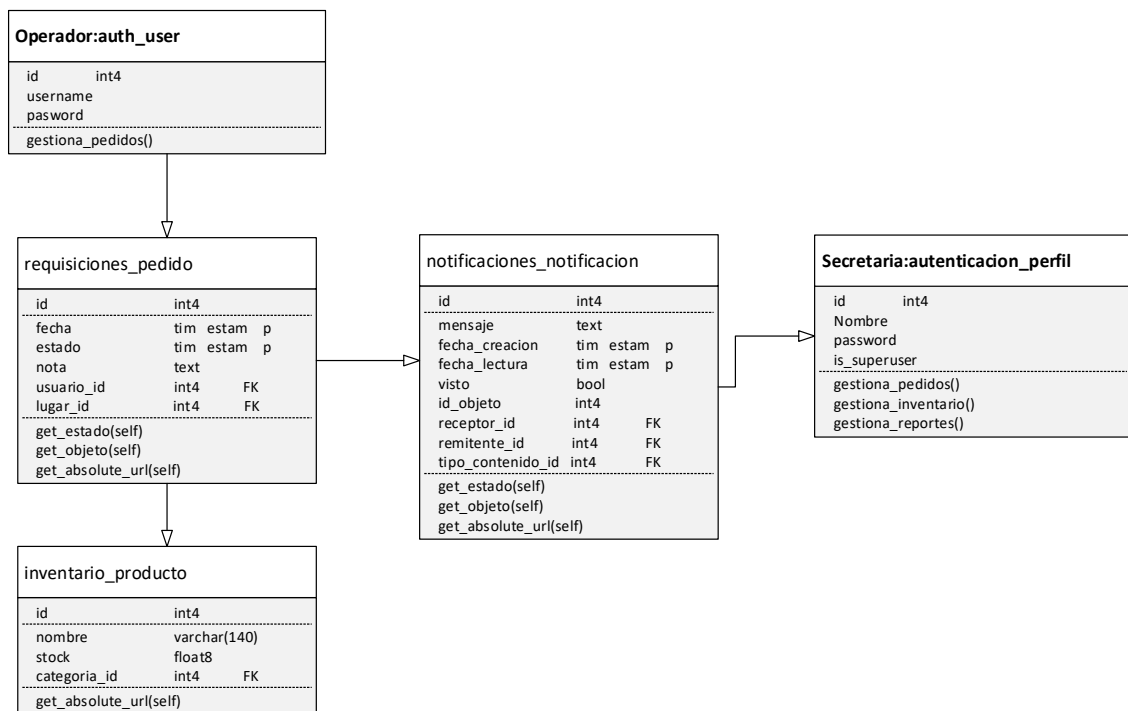


Figura 30-2: Diagrama de objetos – pedido de requisición

Realizado por: Gómez, Diana, 2017.

2.2.2.4 Diseño de interfaces de usuario

Para acceder al sistema de requisiciones, las interfaces gráficas se muestran en un navegador web que permiten la manipulación del sistema por parte del usuario. Al enviar la información a la base de datos se utiliza advertencias y confirmaciones.

En la figura 31-2 se muestra la ventana principal para el logueo de los usuarios.

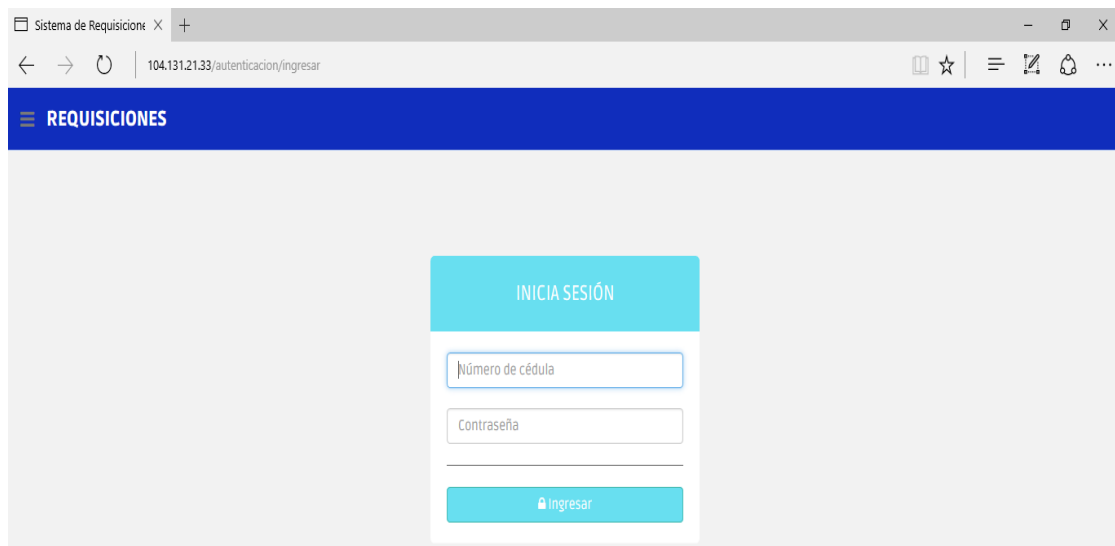


Figura 31-2: Autenticación de usuarios

Realizado por: Gómez, Diana, 2017.

En la figura 32-2 se muestra las opciones bajo el perfil del personal administrativo. El perfil del personal administrativo recibirá notificaciones de requisición, que estarán a la espera de la toma decisiones.

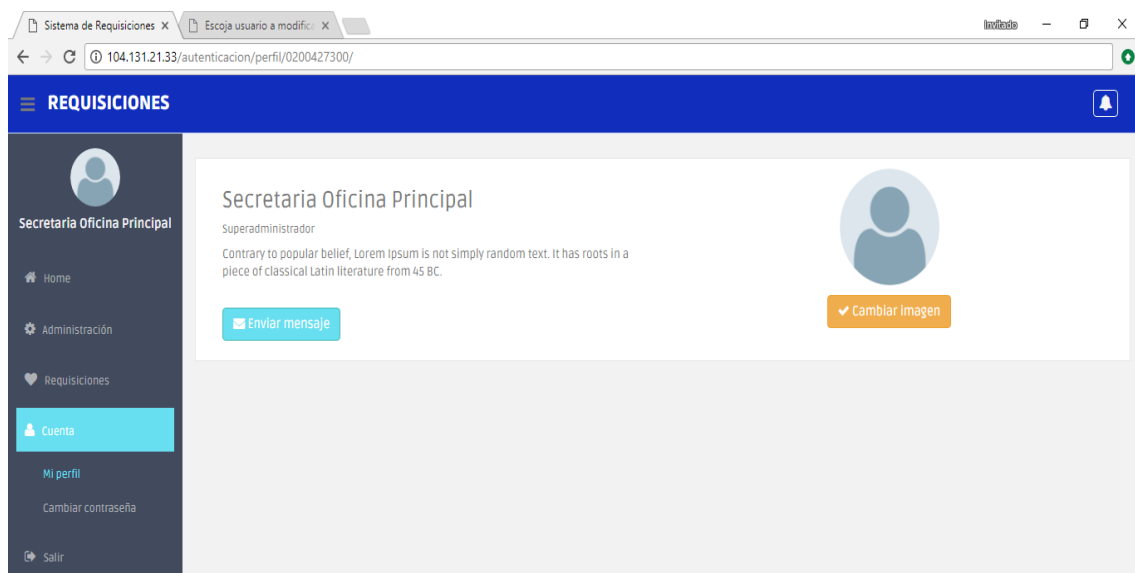


Figura 32-2: Cuenta Administrativo (Secretaria-Gerente)

Realizado por: Gómez, Diana, 2017.

El perfil del personal operativo ingresa los pedidos de requisición, esto se muestra en la figura 33-2.

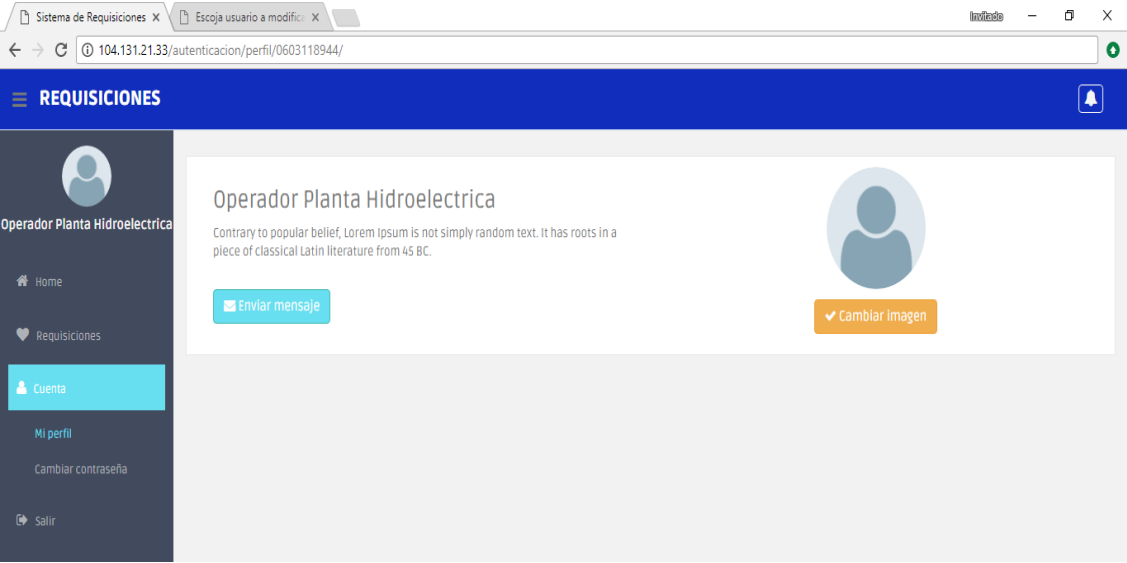


Figura 33-2: Cuenta operativa (Operador)

Realizado por: Gómez, Diana, 2017.

En la figura 34-2 se muestra las opciones por las cuales se pueden generar reportes por parte del perfil del personal administrativo.

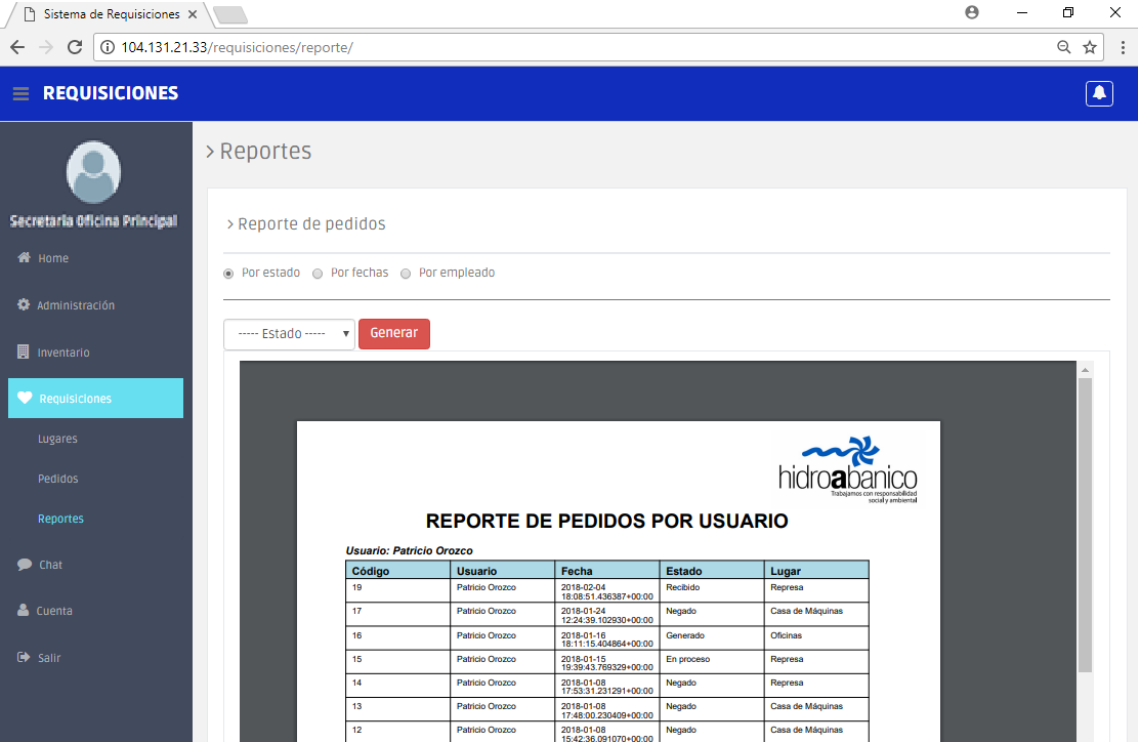


Figura 34-2: Reportes

Realizado por: Gómez, Diana, 2017.

2.2.3 *Diseño Físico*

2.2.3.1 *Diagrama de Implementación*

El diagrama de implementación que indica la figura 35-2 muestra la conexión directa a la base de datos del sistema de requisiciones.

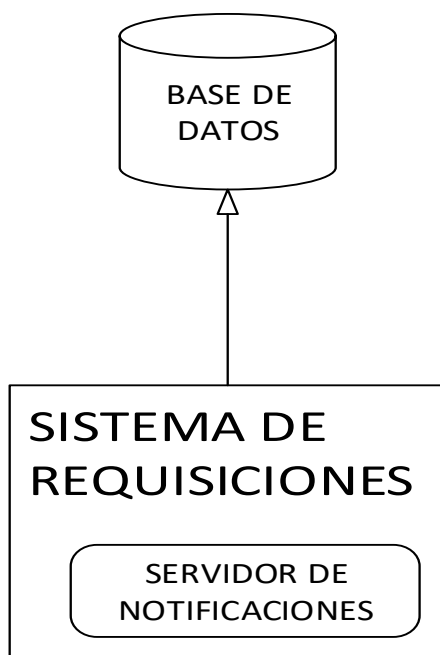


Figura 35-2: Diagrama de implementación

Realizado por: Gómez, Diana, 2017.

2.2.3.2 *Modelo Físico de la Base de Datos*

El modelo físico de la base de datos del sistema de requisiciones se detalla en la figura 36-2.

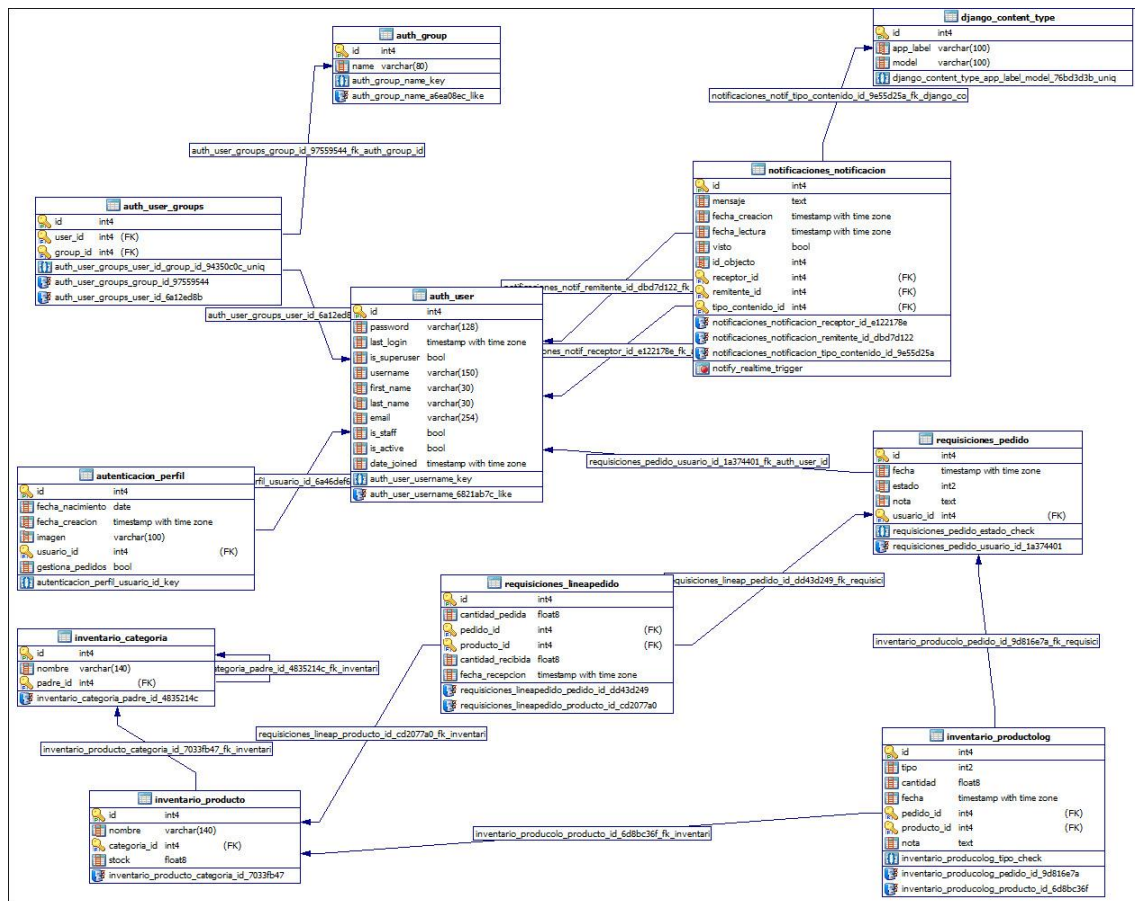


Figura 36-2: Modelo físico base de datos

Realizado por: Gómez, Diana, 2017.

2.3 Desarrollo

2.3.1 Nomenclatura y Estándares

2.3.1.1 Extensiones de Archivos

Los archivos del sistema de requisiciones se detalla la tabla 30-2 donde se explica los formatos y como ser nombrados.

Tabla 26-2: Extensiones Archivos

TIPO DE ARCHIVO	EXTENSIÓN	NOMENCLATURA
templates	.html	nombre_form.html nombre_list.html
páginas dinámicas	.py	nombre.py
scripts	.js	server.js
Imágenes	.jpg .png	nombre.jpg nombre.png
Hojas de Estilo	.css	nombre.css

Realizado por: Gómez, Diana, 2017.

2.3.2 *Recolección y análisis de evidencias primitivas*

La Arquitectura es MVC; Modelo (Model) es la capa de acceso a la base de datos; Vista (Template) esta capa contiene la presentación y Controlador (Views) muestra en la plantilla la lógica que accede al modelo.

2.3.3 *Modelo*

La capa Modelo accede a la base de datos. Contiene la representación de las tablas de la base de datos en forma de clases. El sistema de requisiciones usa el motor PostgreSQL. En la figura 37-2 se puede observar que en cada módulo se encuentra los modelos con el nombre *models.py*.

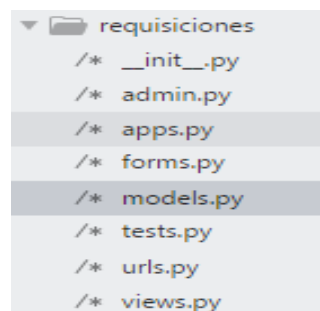


Figura 37-2: *models.py* - Sistema de requisiciones

Realizado por: Gómez, Diana, 2017.

2.3.4 Vista

Los templates se encuentran en la carpeta del mismo nombre, organizados en subcarpetas con los nombres de cada módulo y con los respectivos archivos html.

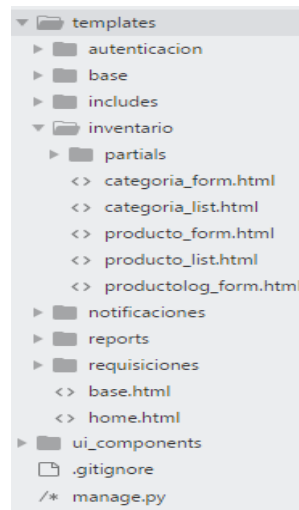


Figura 38-2: Templates

Realizado por: Gómez, Diana, 2017.

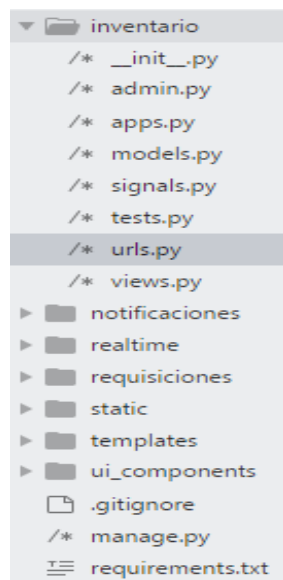


Figura 39-2: Urls.py

Realizado por: Gómez, Diana, 2017.

2.3.5 *Controlador*

Se encuentran en cada módulo con el archivo `views.py` para encapsular la lógica de procesar una petición del usuario y devolver la respuesta.

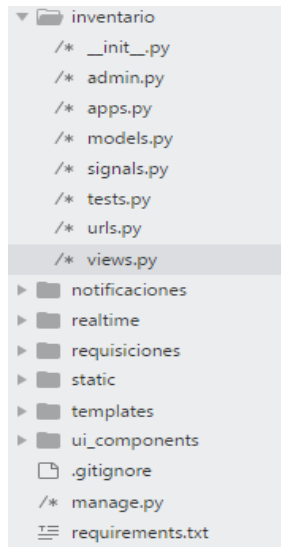


Figura 40-2: Views.py

Realizado por: Gómez, Diana, 2017.

2.4 *Estabilización*

2.4.1 *Revisión del Sistema*

Código Fuente

El desarrollo del sistema de requisiciones está basado en la arquitectura MVC su sigla en español Modelo Vista Controlador.

DESARROLLO DEL SISTEMA DE REQUISICIONES

SERVIDOR DE NOTIFICACIONES

Al ejecutar el siguiente código permite la comunicación en tiempo real entre el personal administrativo y operativo cuando se realiza el pedido de requisición.

```
var express = require('express'),
    app = express();

var server = require('http').createServer(app)
    io = require('socket.io').listen(server);

var pg = require('pg');
var pgConString = "postgres://postgres:081011@localhost:5432/requisicionesdb"
var client = new pg.Client(pgConString)
client.connect()
client.query("LISTEN addnotificacion");
client.handler = false;

var mensaje = require('./db/mensaje');

//mensaje.chat_list(1,3);

var clientes = {}

io.sockets.on('connection', function (socket) {
    if (!client.handler) {
        client.on('notification', function(data) {
            //socket.broadcast.to('gestionadores').emit('message', {data:data.payload});

            var notificacion = JSON.parse(data.payload);
            var receptor = notificacion['receptor_id'];
            if (receptor in clientes) {
                clientes[receptor].socket.emit('count', notificacion)
                clientes[receptor].socket.emit('notificacion', notificacion)
            }
        });
        client.handler=true;
    }

    socket.emit('data', {});

    socket.on('data', function (data) {
        var id = data['user_id']
        this['user'] = data
        clientes[id] = {
            'socket': this,
        }
        socket.emit('ready-chat', {});
    });

    socket.on('chat', (data) => {
        receptor = data['receptor_id'];
        if (receptor in clientes) {
            data['remiteente'] = socket.user;
            clientes[receptor].socket.emit('chat', data);
        }
    });

    socket.on('disconnect', function () {
        delete clientes[this['user']]['user_id'];
        console.log('Usuarios conectados: ' + clientes.length)
        console.log('Socket: ' + this['id'] + ' se ha desconectado.')
    });
});

console.log('Server listen on port 8002');
server.listen(8002)
```

Figura 41-2: Código fuente - Servidor de notificaciones

Realizado por: Gómez, Diana, 2017.

DISEÑO Y CONFIGURACIÓN DE INTERFACES

El diseño y configuración de las interfaces tiene como objetivo que el usuario manipule el sistema de manera fácil e intuitiva. La implementación de los templates se encuentra mediante los templates de django, lo cuales son diseños a base *bootstrap* y *jquery*. El diseño de interfaces se puede observar en el Manual de usuario (**anexo D**).

2.4.2 Plan de pruebas

A continuación del despliegue del sistema de requisición, con la finalidad de determinar si cumple con las perspectivas del usuario y sirve para la depuración de errores. Para esto se ha pedido la intervención de los usuarios como: el personal administrativo (secretaria) de la empresa Hidroabanico S.A.

Posteriormente, se detalla el plan de pruebas planteado para el sistema de requisiciones, mostrando los módulos que fueron puestos a prueba, además del entorno que se utilizó para el proceso.

MÓDULOS A EVALUAR

En el proceso de pruebas para la evaluación se procura aprobar el correcto funcionamiento de los módulos que expone la tabla 31-2.

Tabla 27-2: Componente del sistema evaluado

MÓDULOS	PERSONAL EVALUADOR
Gestión de Usuarios	Administrativo
Gestión de Inventario	Administrativo
Gestión de Requisiciones	Administrativo y Operativo
Gestión de Notificaciones	Administrativo y Operativo
Reportes	Administrativo

Realizado por: Gómez, Diana, 2017.

ENTORNO DE PRUEBAS

Para ejecutar las pruebas se necesitan equipos informáticos con las siguientes características:

Tabla 28-2: Equipos para Pruebas

EQUIPOS CARACTERÍSTICAS	SERVIDOR	CLIENTE
RAM	512 MB	2 GB
DISCO DURO	20 GB	500 GB
SISTEMA OPERATIVO	Ubuntu	Windows
OTROS	Servidor Virtual DigitalOcean	Mozilla Firefox o Google Chrome

Realizado por: Gómez, Diana, 2017.

Las tareas mostradas a continuación, se ejecutan por parte de la secretaria para observar el funcionamiento del sistema:

CREACIÓN DE USUARIOS: crear usuarios y asignar roles.

CREACIÓN DE REQUISICIÓN: crear el pedido de requisición.

GENERACIÓN DE NOTIFICACIONES: las notificaciones sean coherentes con los procesos.

GENERACIÓN DE REPORTES: generar reportes y validar si están acorde a la información generada.

2.5 Soporte

2.5.1 Manuales de Usuario

Es el documento que muestra al usuario como utilizar el sistema de requisiciones. (**anexo D**)

CAPITULO III

3 MARCO DE RESULTADOS

El marco de resultados se basa en la aplicación del estándar ISO/IEC 9126, para la evaluación del producto de origen software teniendo en cuenta las métricas de usabilidad, para esto se tomó el test SUS (System Usability Scale) a los usuarios directos para determinar el grado de usabilidad de la aplicación web.

3.1 Problema

En la empresa Hidroeléctrica Abanico S.A. existen 2 dependencias (represa, casa de máquinas) donde surgen requisiciones de compra, sus operarios elaboran un documento por el cual notifican cuales son los artículos que necesitan. Una vez creado este documento es enviado a través del vehículo institucional hacia la oficina ubicada en el cantón Morona, parroquia Macas. La secretaria recibe este documento, verifica si es posible la compra, realiza la adquisición y envía hacia la dependencia que ha solicitado la requisición. Todo este proceso es lento, tanto al comunicar la necesidad y peor aún en el momento del despacho; además se lo está haciendo de forma manual sin llevar ningún control del mismo.

3.2 Población

La población para tomar el test SUS en base a la métrica de usabilidad en el sistema de requisiciones, serán los empleados de la empresa específicamente 11 personas; dispuestos por el gerente de la empresa Hidroabanico en la provincia de Morona Santiago, quienes van a utilizar la aplicación web.

3.3 Análisis Test SUS

Para evaluar la usabilidad de la aplicación, se realizó la capacitación y uso del sistema web al personal administrativo y operativo de la empresa Hidroeléctrica Abanico S.A., en la cual se puso a consideración el test SUS (**anexo E**). Una vez completadas las encuestas a los empleados se procede a la tabulación de las mismas. Cabe recalcar que la escala de usabilidad va desde completamente de acuerdo (5) a completamente desacuerdo (1).

En la tabla 33-3 se realizó la tabulación de los datos obtenidos en las encuestas realizadas a los empleados, sobre el uso del sistema de requisiciones implementado en la empresa. Con los siguientes resultados:

Tabla 29-3: Tabulación datos Test SUS

N°	PREGUNTA	N° de test											
		(# Usuarios)											
		1	2	3	4	5	6	7	8	9	10	11	
1	Creo que me gustaría utilizar este sistema frecuentemente.	5	4	5	5	5	4	4	5	5	4	5	92,73
2	El sistema me resulto innecesariamente complejo.	2	2	1	1	1	1	2	2	1	2	1	29,09
3	Creo que el sistema es bastante fácil de utilizar.	5	5	5	5	5	5	5	5	5	5	5	100,00
4	Creo que necesitaría el soporte de un técnico para poder utilizar este sistema.	1	1	1	2	1	1	1	1	1	1	2	23,64
5	Creo que as diferentes funciones del sistema se encuentran muy bien integradas.	5	5	5	5	4	5	4	5	5	5	5	96,36
6	Opino que hubo demasiada inconsistencia en el sistema.	1	2	1	1	1	1	1	1	2	1	1	23,64
7	Imagino que la mayoría de las personas aprendería a utilizar el sistema rápidamente.	4	5	5	5	4	5	5	5	4	5	5	94,55
8	Me sentí algo incómodo al utilizar este sistema.	1	1	1	1	1	2	1	1	1	1	1	21,82
9	Me sentí muy seguro al utilizar este sistema.	5	5	5	4	4	4	5	5	5	5	5	94,55
10	Necesito aprender muchas otras cosas antes de poder utilizar correctamente el sistema.	2	2	1	2	2	2	1	1	2	2	1	32,73
TOTAL		37	36	40	37	36	36	37	39	37	37	39	93,41
Puntuación SUS		93	90	100	93	90	90	93	98	93	93	98	93,41

Realizado por: Gómez, Diana, 2018.

Análisis: Se realizó la encuesta a 11 empleados de la empresa, en la cual se obtuvo los siguientes porcentajes de usabilidad; el usuario 1 obtiene un 93%, el usuario 2 obtiene un 90%, el usuario 3 obtiene un 100%, el usuario 4 obtiene un 93%, el usuario 5 obtiene un 90%, el usuario 6 obtiene un 90%, el usuario 7 obtiene un 93%, el usuario 8 obtiene un 98%, el usuario 9 obtiene un 93%, el usuario 10 obtiene un 93%, el usuario 11 obtiene un 98%. En promedio el sistema de requisiciones obtiene 93.41% de aceptación por parte de los usuarios luego de ser aplicado el test SUS (System Usability Scale).

En la Tabla 33-3 se obtuvo un promedio de los resultados en cada pregunta del test SUS, con lo cual se realizó un gráfico estadístico de la incidencia de cada pregunta realizada.

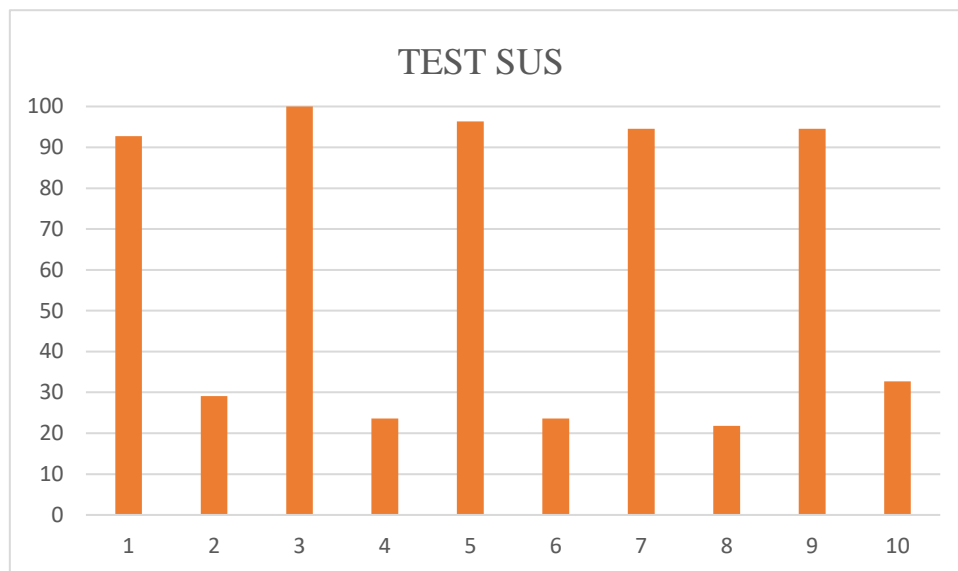


Gráfico 1-3: Evaluación por pregunta
Realizado por: Gómez, Diana, 2018.

Análisis: En el gráfico 1-3 se observa el porcentaje obtenido en cada pregunta. En las preguntas impares se obtuvo los siguientes resultados: pregunta 1 = 92.73%, pregunta 3 = 100%, pregunta 5 = 96.36%, pregunta 7 = 94.55%, pregunta 9 = 94.55%; cabe recalcar que según el test SUS estas preguntas son consideradas como positivas porque su cercanía al completamente de acuerdo (5).

En las preguntas pares se obtuvo los siguientes resultados: pregunta 2 = 29.09%, pregunta 4 = 23.64%, pregunta 6 = 23.64%, pregunta 8 = 21.82%, pregunta 10 = 32.73%; según el test SUS estas preguntas son consideradas como negativas por su cercanía al completamente desacuerdo (1).

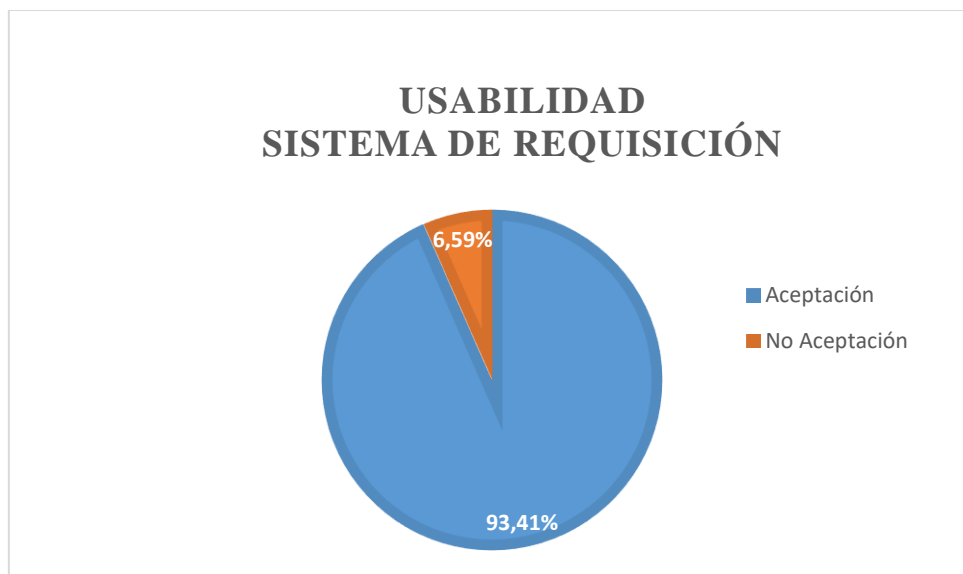


Gráfico 2-3: Usabilidad Sistema de Requisición

Realizado por: Gómez, Diana, 2018.

Análisis: En el gráfico 2-3 se observa el grado de usabilidad que obtuvo el sistema de requisiciones implementado en la empresa Hidroabánico S.A., que es del 93,41% por lo que el sistema cumple con los parámetros básicos de usabilidad para el usuario.

3.4 Análisis de resultados

Para iniciar el desarrollo de la investigación, se obtuvo información mediante la entrevista realizada al personal administrativo de la empresa. A continuación, se pasó a la etapa de análisis donde, se definió los requerimientos funcionales y no funcionales. Para el desarrollo de la aplicación se aplicó la metodología MSF (Microsoft Solutions Framework) que resulta ser un método estructurado, flexible y eficiente para un grupo de desarrollo pequeño.

Una vez concluido el proceso de desarrollo de la aplicación, se procede a realizar el test SUS (System Usability Scale) a los usuarios reales de prueba, además de realizar pruebas reales en el sistema con los usuarios.

Al termino del desarrollo de la aplicación de requisiciones, su resultado permite ver que:

- La aplicación web nos ayuda con disponibilidad en cuanto tiempo y lugar, se podrá trabajar en ella en el momento que surge el pedido de requisición y en cualquiera de sus plantas u oficina mientras tengan conexión a internet.
- Para trabajar en la aplicación de requisiciones solo necesita un computador, un navegador web y conexión a internet.
- La aplicación web sirve para mantener una comunicación adecuada continua entre el personal operativo que se encuentra en las plantas (casa de máquinas o represa) y el personal administrativo.
- La utilización de la aplicación por parte del personal administrativo y operativo, no requiere una guía constante por parte del desarrollador ya que la página web es de fácil interpretación para que el usuario, cabe recalcar que los usuarios deberán tener un mínimo de conocimientos en la utilización del computador.
- Una vez aplicado el test SUS, una encuesta estandarizada de 10 preguntas para medir la usabilidad de un producto software, la tabulación arroja el 93.41% en la escala de usabilidad.

CONCLUSIONES

1. La utilización del sistema de requisiciones permite incrementar la comunicación entre los empleados a través de notificaciones en tiempo real, además agiliza la toma de decisiones por parte del personal administrativo, quien se basará en el inventario o reportes que proporciona la página web.
2. La utilización de javascript del lado del servidor a través de node.js permite la creación del sistema web de requisiciones en tiempo real a través de websockets, que no permite puntos muertos ni bloqueos por llamadas de entrada y salida, basada en la comunicación cliente\servidor.
3. El framework de desarrollo de sitios web Django permite protección a las cuentas de usuario y contraseñas, evitando errores comunes como colocar la información de la sesión en cookies donde es vulnerable, protege la inyección SQL; además permite la reutilización de componentes sin la necesidad de empezar desde cero.
4. La metodología de desarrollo Microsoft Solutions Framework permite trabajar en grupo reducido de desarrollo, además soporta adaptabilidad a cambios repentinos en el sistema sin afectar el trascurso de la ingeniería.
5. Al aplicar el test SUS (System Usability Scale) basado en la Norma ISO/IEC 9126-1 para medir la usabilidad del sistema de requisiciones, se obtuvo un 93.41%, lo cual indica la aceptación por parte de los usuarios hacia la aplicación web.

RECOMENDACIONES

1. Para que la aplicación muestre resultados adecuados, es recomendable que las creaciones de cuenta de empleados sean asignados a un solo personal administrativo, en este caso a la secretaria para que a futuro cuando se realice los reportes no exista inconsistencia en los datos.
2. Para el desarrollo de aplicaciones web que necesiten comunicación en tiempo real se recomienda el uso del framework *NodeJS* ya que este implementa librerías para trabajar con websockets que permiten la comunicación inmediata entre el cliente y servidor.
3. Es recomendable utilizar el framework *Django* porque ofrece la reutilización de componentes software sin necesidad de empezar de cero, acortando el tiempo de desarrollo de una aplicación web.
4. La metodología MSF (*Microsoft Solutions Framework*) abarca las etapas del desarrollo desde el análisis de los requerimientos, gestión de riesgos hasta las pruebas e implementación lo cual permitió que el sistema de requisiciones cumpla con las expectativas propuestas al usuario, por ende, se recomienda para el desarrollo rápido de aplicaciones web.
5. Se recomienda el uso del test SUS (*System Usability Scale*) para medir la usabilidad de aplicaciones ya que de forma sencilla y con 10 preguntas se obtienen resultados muy aceptables.
6. Se recomienda a futuro que el sistema puede incluir el módulo de proveedores, ya que este proceso se lo maneja de una forma aislada debido a que el sistema desarrollado solo se basa en la notificación de las requisiciones.

BIBLIOGRAFÍA

- **CASTRELO, Alejandro.** MMO de Navegador en Tiempo Real con Node.js y WebSockets. (Tesis de Ingeniería Informática). [en línea] Universidad de Barcelona, Facultad de Matemáticas, Barcelona, 2014. [Consulta: 14 de marzo 2018]. Disponible en: <http://diposit.ub.edu/dspace/bitstream/2445/59452/1/memoria.pdf>
- **CATALDI, Zulma.** Metodología de diseño, desarrollo y evaluación de software educativo. (Tesis de maestría). [en línea] Universidad Nacional de la Plata, Facultad de Informática, Colombia. 2000. [Consulta: 10 de enero 2018]. Disponible en: http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/2737/0053L864e_anexo.pdf;jsessionid=49CEC8600FD92C87B4599BBE4FAEAD6A?sequence=2
- **CONDORI, José.** “Phython – Django. Framework de desarrollo web para perfeccionistas. Basado en el Modelo MTV” [en línea]. *Revista de Información, Tecnología y Sociedad*, n°7 (2012), (Bolivia) pp. 36-37. Disponible en: http://www.revistasbolivianas.org.bo/scielo.php?script=sci_arttext&pid=S1997-40442012000200016&lng=es&nrm=iso
- **DJANGO SOFTWARE FOUNDATION.** *Django Documentation* [en línea]. Writing your first Django app, part 5. Disponible en: <https://docs.djangoproject.com/en/1.11/intro/tutorial05/>
- **DRAKE, Fred.** *El tutorial de Python* [en línea]. Argentina, Python Software Foundation: 2017, pp.1-3-8-9-11-12-13-66. [Consulta: 5 febrero 2017] Disponible en: <http://docs.python.org.ar/tutorial/pdfs/TutorialPython3.pdf>
- **DURÁN, Cristina.** La requisición de la industria durante la unidad popular. Relación a un concepto. [en línea]. Santiago-Chile: Universidad de Chile. [Consulta: 8 de enero 2018]. Disponible en: < <http://repositorio.uchile.cl/bitstream/handle/2250/139941/La-requisici%C3%B3n-de-la-industria-durante-la-Unidad-Popular.pdf?sequence=1> >
- **FLORES, Ángel.** Desarrollo de un Entorno Virtual Colaborativo Aplicado a la Enseñanza del Diseño Web, en la Epoch Extensión Morona Santiago para mejorar el aprendizaje significativo (Tesis de maestría). [En línea] Escuela Superior Politécnica de

Chimborazo, Facultad de Informática y Electrónica. Riobamba-Ecuador. 2016.
[Consulta: 10 febrero 2018]. Disponible en:

<http://dspace.esPOCH.edu.ec/bitstream/123456789/4687/1/20T00704.pdf>

- **FLORÍA, Alejandro.** *Fundación sidar* [en línea]. 2000. Métodos de Indagación: Algunos cuestionarios conocidos. [Consulta: 10 febrero 2018] Disponible en: <https://www.sidar.org/recur/desdi/traduc/es/visitable/nuevos/CuestCon.htm>
- **GARCÍA, Ana.** *Evaluación de métricas de calidad del software sobre un programa Java (Tesis) (maestría)*. Universidad Complutense de Madrid, Facultad de Informática, Departamento de Sistemas Informáticos y Computación. Madrid-España. 2010. pp. 7-8 [Consulta: 15 julio 2007]. Disponible en: <http://eprints.ucm.es/11487/>
- **GARCÍA, Dina.** Propuesta para mejorar la respuesta a las requisiciones de compra de crear ingeniería Ltda. Mediante simulación en arena y prueba bonferroni. (Trabajo Aplicado). [en línea]. Politécnico Grancolombiano Institución Universitaria, Facultad de Ingeniería Industrial, Especialización en logística y gestión de la cadena de abastecimiento. Bogotá- Colombia. 2015. [Consulta: 15 de marzo 2018]. Disponible en: <http://repository.poligran.edu.co/bitstream/handle/10823/648/Proyecto%20de%20Grado%20Antonia%20Garcia.pdf?sequence=1&isAllowed=y>
- **GARCÍA, Oscar.** Northware. Software Development. [en línea]. 2014. Aplicaciones en Tiempo Real con node.js. [Consulta: 14 de marzo del 2018]. Disponible en: http://www.northware.mx/wp-content/uploads/2014/03/northware_marzo2014_aplicaciones_tiempreal_nodejs.pdf?x69168
- **GAUCHAT, Juan.** EL GRAN LIBRO DE HTML5, CSS3 Y JAVASCRIPT, Primera Edición, Barcelona-España, Editorial MARCOMBO: 2012, pp. 25-70.
- **GÓMEZ Andrés., & ALVARADO, Laura.** Análisis de usabilidad de la página web de la Universidad Católica de Colombia. (Trabajo de Investigación). [en línea]. Universidad Católica de Colombia, Facultad de Ingeniería en Sistemas. Bogotá-Colombia. 2015. [Consulta: 15 de marzo 2018]. Disponible en: <http://repository.ucatolica.edu.co/bitstream/10983/1295/1/TRABAJO%20DE%20GRADO.pdf>

- **GUADALUPE, Eduardo., & MOROCHO, Saúl.** Desarrollo de una Aplicación Web Utilizando la Norma NTE INEN-ISO/IEC 9126-1 para Automatizar el Cobro del Servicio de Agua Potable en la Provincia de Morona Santiago (Tesis de pregrado). [en línea] Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica. Riobamba-Ecuador. 2016. [Consulta: 15 agosto 2017]. Disponible en: <http://dspace.espoch.edu.ec/handle/123456789/4732#sthash.iIHj1XJS.dpuf>
- **INFANTE, Sergio.** *Maestros del Web* [en línea]. 2012. Curso Django: Entendiendo como Funciona Django. [Consulta: 14 febrero 2017] Disponible en: <http://www.maestrosdelweb.com/curso-django-entendiendo-como-trabaja-django>
- **LANDÁZURI, Carlos.** Estudio de la metodología MSF “Microsoft Solutions Framework” aplicada al desarrollo de un sistema de facturación “software libre “para la unión de papelerías de la ciudad de Ibarra. [en línea]. Ecuador- Ibarra: Universidad Técnica del Norte. [Consulta: 14 de marzo 2018]. Disponible en: <http://repositorio.utn.edu.ec/bitstream/123456789/1830/3/ISC%20258%20Art%C3%A1culo%20Cient%C3%ADfico%20Español.pdf>
- **MICROSOFT SOLUTIONS FRAMEWORK (MSF).** *Microsoft Documentation* [en línea]. Microsoft solutions Framework (MSF) Overview. Disponible en: [https://msdn.microsoft.com/es-es/library/jj161047\(v=vs.120\).aspx](https://msdn.microsoft.com/es-es/library/jj161047(v=vs.120).aspx)
- **MOZILLA, mdn.** Acerca de Javascript. [en línea]. [Consulta: 14 de marzo 2018]. Disponible en: https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript
- **NOVAK, Fabián.** La contratación entre estados y empresas transnacionales. [en línea]. Perú: Pontificia Universidad Católica del Perú. [Consulta: 8 de enero 2018] Disponible en: <http://revistas.pucp.edu.pe/index.php/agendainternacional/article/viewFile/7160/7360>
- **ONGEI, Mark.** Guía Técnica sobre Evaluación de software en la Administración Pública [en línea]. Piura-Perú: Oficina Nacional de Gobierno Electrónico e Informática. [Consulta: 15 de marzo 2017]. Disponible en: http://www.ongei.gob.pe/bancos/banco_normas/archivos/guia-evaluacion-sw.pdf

- **ROSSUM, Guido van.** Guía de aprendizaje de Python. [en línea]. BeOpen PythonLabs. 2015. [Consulta: 15 de marzo 2018]`. Disponible en: <http://es.tldp.org/Tutoriales/Python/tut.pdf>
- **SUMMERFIELD, Mark.** *Python3*. Madrid-España: GRUPO ANAYA: 2010, p.1
- *Usabilidad web, un test que te dará grandes resultados* [en línea]. QuestionPro. 2018. [Consulta: 24 de febrero 2018]. Disponible en: <https://www.questionpro.com/blog/es/usabilidad-web-test/>
- **WORSLEY, John; & DRAKE, Josua.** *Practical PostgreSQL*. Sebastopol, CA-Estados Unidos: O'Reilly & Associates Inc, 2002, p 3.

GLOSARIO

A

Aplicación web: sistema desarrollado de tal forma que funcione desde la red, es decir, accediendo a un servidor web mediante un navegador.

C

Caso de uso: Es un diagrama o procedimiento realizado en la fase de ingeniería para desempeñar dentro del sistema determinada función.

D

Django: Framework de desarrollo de páginas web.

E

Entorno: aplicación informática que facilita el desarrollo del software.

F

Framework: Estructura que sirve de base para la organización y desarrollo de software.

H

Hidroeléctrica: Empresa dedicada a la toma de agua para generar energía eléctrica.

Hidroabánico: Hidroeléctrica que usa el agua del río Abanico ubicado en la provincia de Morona Santiago.

I

Interfaz: Es un vínculo o conexión funcional entre el sistema y el usuario para el intercambio de información.

J

Javascript: Lenguaje de programación orientado a objetos.

L

Librería: Conjunto de implementaciones que desarrollan una funcionalidad a la que se invoca, codificados en un lenguaje de programación

M

Modelo: Representación de los datos que maneja el sistema.

N

Notificación: documento o acción de comunicar algo.

P

Python: Lenguaje de programación de código abierto.

R

Requisición: Documento con cual se notifica la necesidad de compra de un bien o servicio.

S

Socket: Interfaz de programación de aplicaciones para los protocolos TCP/IP.

T

Template: Diseño web que define el esqueleto del programa.

V

Vista: Interfaz de usuario

A N E X O S

ANEXO A

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

Encuesta para el desarrollo de un sistema para mejorar la notificación de pedido de requisiciones.

El siguiente cuestionario pretende evaluar el grado de eficiencia en base a tiempos del proceso de notificación de requisición entre el operador y la secretaria.

Las opciones en las siguientes preguntas en base a las respuestas representan:

a. Muy malo	b. Malo	c. Regular	d. Bueno	e. Muy Bueno
-------------	---------	------------	----------	--------------

Indicación: Por favor conteste el presente cuestionario según su criterio.

Cargo:

Género: () Masculino () Femenino

1. ¿Cuál es el tiempo que le toma en notificar un pedido de requisición?

- a) 8 horas o más
- b) De 4 a 8 horas
- c) De 2 a 4 horas
- d) De 1 a 2 horas
- e) Menos de 1 hora

¿Explique por qué?

2. ¿Cuál es el tiempo que le toma en notificar si fue aceptado o rechazado el pedido de requisición?

- a) 8 horas o más
- b) De 4 a 8 horas
- c) De 2 a 4 horas
- d) De 1 a 2 horas
- e) Menos de 1 hora

¿Explique por qué?

3. Después de haber realizado el pedido de requisición. ¿Cuál es el tiempo que tarda en procesar el pedido de requisición?

- a) 8 horas o más
- b) De 4 a 8 horas
- c) De 2 a 4 horas
- d) De 1 a 2 horas
- e) Menos de 1 hora

¿Explique por qué?

4. Después de haber finalizado el proceso del pedido de requisición. ¿Cuál es el tiempo que tarda en realizar un reporte de seguimiento al pedido de requisición?

- f) 8 horas o más
- g) De 4 a 8 horas
- h) De 2 a 4 horas
- i) De 1 a 2 horas
- j) Menos de 1 hora

¿Explique por qué?

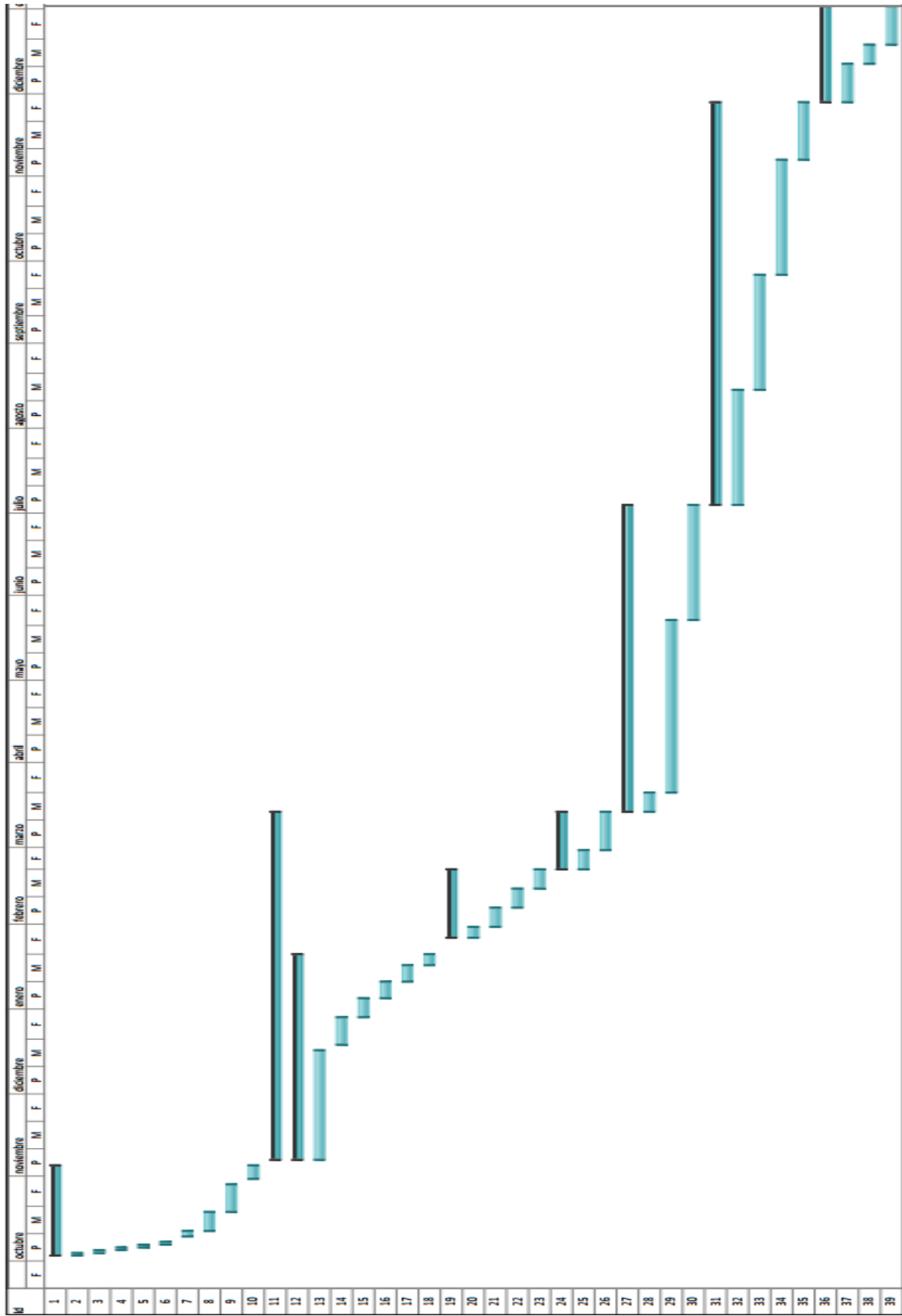
ANEXO B

DIAGRAMA GANTT Y DIAGRAMA DE SEGUIMIENTO

DIAGRAMA GANTT

	Nombre de tarea	Duración	Comienzo	Fin
Diagrama de Gantt	1 - VISION Y ALCANCE	25 días	lun 03/10/16	vie 04/11/16
	Definición del Problema	1 día	lun 03/10/16	lun 03/10/16
	Visión del Proyecto	1 día	mar 04/10/16	mar 04/10/16
	Perfiles de Usuario	1 día	mié 05/10/16	mié 05/10/16
	Ambito del Proyecto	1 día	jue 06/10/16	jue 06/10/16
	Herramientas a Utilizar	1 día	vie 07/10/16	vie 07/10/16
	Objetivos del Proyecto	2 días	lun 10/10/16	mar 11/10/16
	Análisis y Gestión del Riesgo	5 días	mié 12/10/16	mar 18/10/16
	Planificación Inicial	8 días	mié 19/10/16	vie 28/10/16
	Estudio de Factibilidad	5 días	lun 31/10/16	vie 04/11/16
Diagrama de Gantt	2 - PLANIFICACIÓN	91 días	lun 07/11/16	lun 13/03/17
	2.1 DISEÑO CONCEPTUAL	55 días	lun 07/11/16	vie 20/01/17
	Interfaces	30 días	lun 07/11/16	vie 16/12/16
	Requerimientos Funcionales	8 días	lun 19/12/16	mié 28/12/16
	Requerimientos No Funcionales	5 días	jue 29/12/16	mié 04/01/17
	Actores	4 días	jue 05/01/17	mar 10/01/17
	Casos de Uso	4 días	mié 11/01/17	lun 16/01/17
	Arquitectura Apropiada para la solución	4 días	mar 17/01/17	vie 20/01/17
	2.2 DISEÑO LÓGICO	17 días	vie 27/01/17	lun 20/02/17
	Arquitectura a utilizar en el proyecto	2 días	vie 27/01/17	lun 30/01/17
	Diagramas de Secuencia	5 días	mar 31/01/17	lun 06/02/17
	Diagramas de Clases	5 días	mar 07/02/17	lun 13/02/17
	Diseño de Interfaces de Usuario	5 días	mar 14/02/17	lun 20/02/17
	2.3 DISEÑO FISICO	15 días	mar 21/02/17	lun 13/03/17
	Diagrama de Implementación	5 días	mar 21/02/17	lun 27/02/17
	Modelo Físico de la Base de Datos	10 días	mar 28/02/17	lun 13/03/17
	3 - DESARROLLO	80 días	mar 14/03/17	lun 03/07/17
	Nomenclatura y Estándares	5 días	mar 14/03/17	lun 20/03/17
	Modelo Vista Template	45 días	mar 21/03/17	lun 22/05/17
	Servidor de Notificaciones	30 días	mar 23/05/17	lun 03/07/17
Diagrama de Gantt	4 - ESTABILIZACIÓN	105 días	mar 04/07/17	lun 27/11/17
	Revisión General del Sistema	30 días	mar 04/07/17	lun 14/08/17
	Pruebas	30 días	mar 15/08/17	lun 25/09/17
	Correcciones	30 días	mar 26/09/17	lun 06/11/17
	Pruebas	15 días	mar 07/11/17	lun 27/11/17
	5 - INSTALACIÓN	25 días	mar 28/11/17	lun 01/01/18
	Plan de Instalación y Soporte	10 días	mar 28/11/17	lun 11/12/17
	Manuales de Usuario	5 días	mar 12/12/17	lun 18/12/17
	Revisión Documento Final	10 días	mar 19/12/17	lun 01/01/18

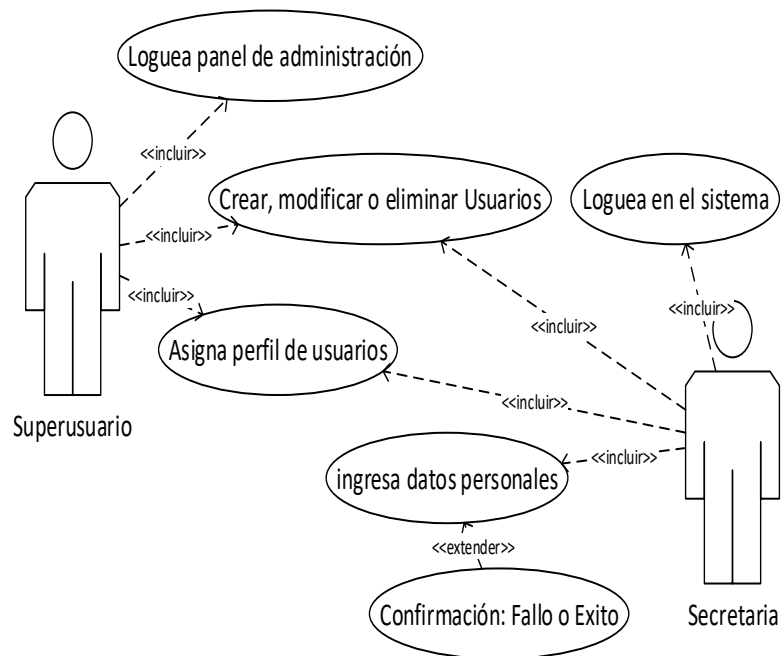
DIAGRAMA GANTT DE SEGUIMIENTO



ANEXO C

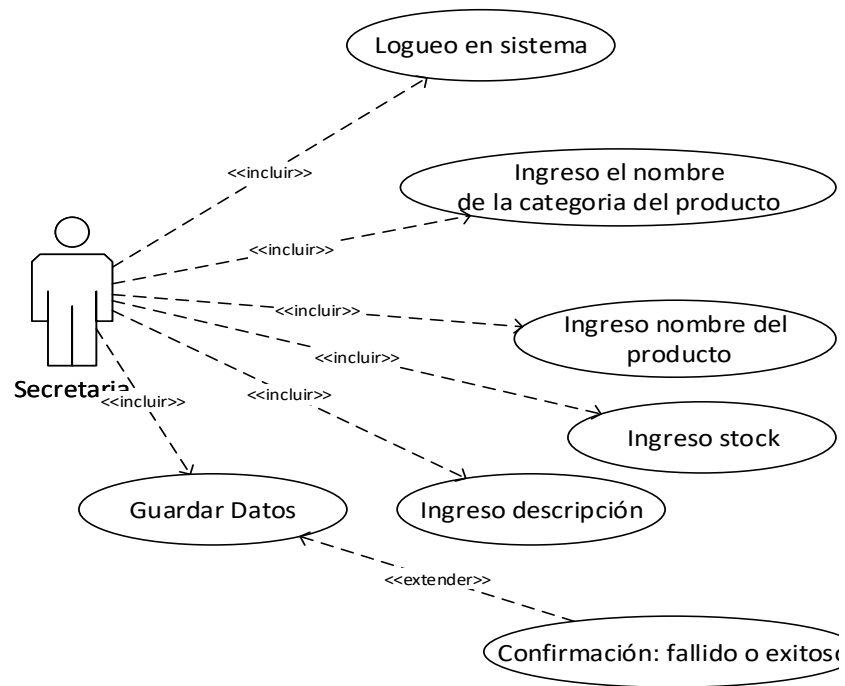
CASOS DE USO

REQUERIMIENTO # 1



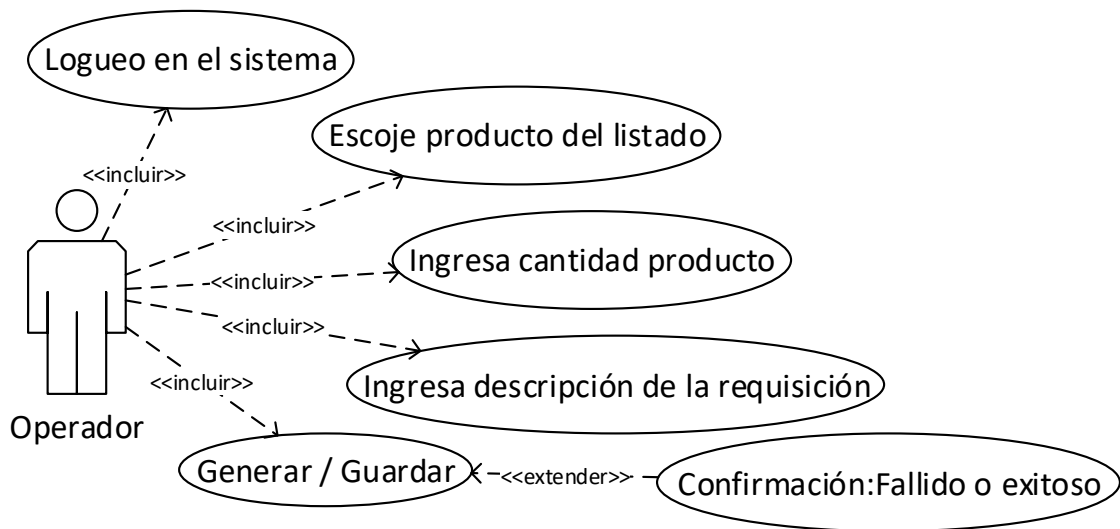
Número: Requerimiento 1	Usuario: Administrativo
Nombre historia: Gestión de usuarios	
Responsable: Diana E. Gómez G.	
Descripción: El módulo de gestión de usuarios es administrado bajo el perfil de superusuario o administrativo donde se creará usuario, contraseña y se asigna su perfil de usuario..	
ACCIONES DE ACTORES	RESPUESTA DEL SISTEMA
Ingresa al sistema	Valida perfil de usuario.
Crear, modificar o eliminar usuarios.	Ejecuta acción
Ingreso cédula y contraseña.	Valida campos, guarda información en base de datos
Ingreso la información requerida para el registro del usuario.	Valida campos, guarda información en base de datos
asignar perfil a los usuarios.	Ejecuta acción
Observaciones: Usuario ingresa información incorrecta el sistema informa y no ejecuta la acción.	

REQUERIMIENTO # 2



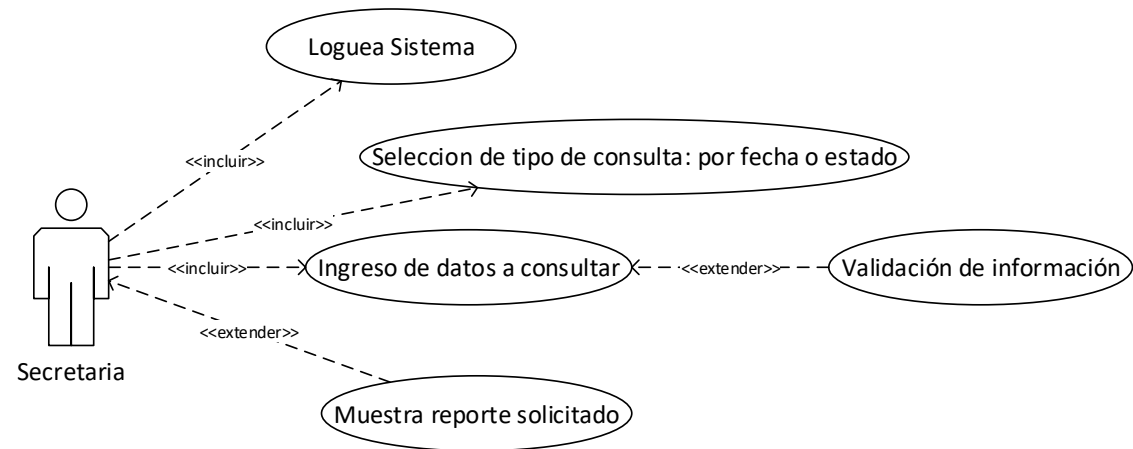
Número: Requerimiento 2	Usuario: Superusuario - Administrativo
Nombre historia: Gestión de Inventario	
Responsable: Diana E. Gómez G.	
Descripción: El módulo de inventario es administrado bajo el perfil de superusuario o administrativo donde se ingresa, modifica o elimina productos o categorías.	
ACCIONES DE ACTORES	RESPUESTA DEL SISTEMA
Ingresa al sistema	Valida perfil de usuario.
crear, modificar o eliminar categorías.	Ejecuta acción
Ingreso información requerida para registro de la categoría.	Valida campos, guarda información en base de datos
crear, modificar o eliminar productos.	Ejecuta acción
Ingreso información requerida para registro del producto.	Valida campos, guarda información en base de datos
Crear, modificar o eliminar stock.	Ejecuta acción.
Registro información para el stock.	Valida campos, guarda información en base de datos
Observaciones: Usuario ingresa información incorrecta el sistema informa y no ejecuta la acción.	

REQUERIMIENTO # 3



Número: Requerimiento 3	Usuario: Operador - Administrativo
Nombre historia: Gestión de Requisiciones	
Responsable: Diana E. Gómez G.	
Descripción: El módulo de requisiciones es administrado bajo el perfil operativo o administrativo donde se ingresa, modifica o elimina pedidos de requisición.	
ACCIONES DE ACTORES	RESPUESTA DEL SISTEMA
Ingresa al sistema	Valida perfil de usuario.
Elegir los productos del inventario.	Ejecuta acción
Ingreso cantidad del producto solicitado	Verifica cantidad en stock
Ingreso descripción de la requisición.	Valida campos, guarda información en base de datos.
Observaciones: Usuario ingresa información incorrecta el sistema informa y no ejecuta la acción	

REQUERIMIENTO # 5



Número: Requerimiento 5	Usuario: Secretaria
Nombre historia: Gestión de reportes	
Programador responsable: Diana E. Gómez G.	
Descripción: El sistema permite la visualización de reportes.	
ACCIONES DE ACTORES	RESPUESTA DEL SISTEMA
Ingresa al sistema	Valida perfil de usuario.
Selecciona estado, fecha o empleado	Ejecuta acción
Observaciones: Usuario ingresa información incorrecta el sistema informa y no ejecuta la acción	

ANEXOD

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERIA EN SISTEMAS



MANUAL DE USUARIO

SISTEMA DE REQUISICIONES HIDROABANICO S.A.

MACAS – ECUADOR

2018

CONTENIDO

INTRODUCCIÓN	12
REQUISITOS PARA EL USO DEL SISTEMA	12
CATEGORÍAS DE USUARIO	13
<input type="checkbox"/> Administrativo o Administrador.....	13
<input type="checkbox"/> Operativo	13
INGRESO AL SISTEMA.....	14
OPCIONES DE MENÚ	15
1. ADMINISTRACIÓN	15
Creación Cuenta de Usuario	15
2. INVENTARIO	16
CATEGORIAS	17
Ingreso Listado Categorías	17
Creación de Nueva Categoría	18
PRODUCTOS.....	19
Ingreso Listado Productos	19
Creación Nuevo Producto	19
Ingreso o Editar de Datos del Producto	20
Eliminar producto	21
3. REQUISICIONES.....	22
LUGARES	22
Ingreso Listado Lugares.....	22
Creación Lugar	23
Eliminar Lugar.....	23
PEDIDOS 24	
Creación de Pedido	24
Procesamiento del Pedido	25
REPORTES	27
Creación de Reportes	27
Reporte por estado	28
Reporte por fecha.....	28
Reporte por empleado.....	29
4. CHAT	30
5. CUENTA.....	30
Acceso a perfil de usuario	30
Cambio de Contraseña	31
6. SALIR	31

INTRODUCCIÓN

El siguiente documento llamado *Manual De Usuario* sirve para desempeñar correctamente el sistema de requisiciones desarrollado para la empresa Hidroabanico S.A. Esta guía esta explicada de una manera sencilla y detallada para ser un instrumento de orientación sobre las funcionalidades del sistema.

Es recomendado que el usuario revise esta guía para que se familiarice con el sistema de requisiciones y sus funciones.

El sistema de requisiciones es una aplicación web basada en la comunicación a través de notificaciones entre los departamentos de la empresa para la elaboración de pedidos de requisición. Existe dos roles para acceder a la aplicación: Administrativo o Administrador y Operativo.

REQUISITOS PARA EL USO DEL SISTEMA

- Un computador de escritorio o laptop.
- Conexión a internet
- Tener instalado un navegador web (Mozilla Firefox o Google Chrome).

CATEGORÍAS DE USUARIO

- **Administrativo o Administrador.** – Este perfil tiene acceso total a todas las opciones del menú.



Ilustración 1: Opciones de Menú - Usuario Administrativo

- **Operativo.** - Este perfil tiene acceso al menú requisiciones.



Ilustración 2: Opciones de Menú - Usuario Operativo

INGRESO AL SISTEMA

Para acceder a la aplicación, el usuario debe seguir los siguientes pasos:

- Digitar en el navegador web el URL: 104.131.21.33
- Ingresar sus credenciales de acceso (Número de cédula y contraseña).
- Clic en Ingresar

Importante

En el caso de que el usuario no tenga credenciales de acceso, debe ponerse en contacto con el usuario Administrativo a cargo de la empresa para solicitarle las respectivas credenciales.



The image shows a web browser window with the title 'Sistema de Requisiciones'. The address bar shows the URL '104.131.21.33/autenticacion/ingresar/' and a 'No seguro' (Not secure) warning. The page has a blue header with the text 'REQUISICIONES' and a hamburger menu icon. The main content area features a light blue box with the title 'INICIA SESIÓN'. Inside this box, there are two input fields: 'Número de cédula' and 'Contraseña'. Below these fields is a red button with a lock icon and the text 'Ingresar'.

Ilustración 3: Inicio de Sesión

OPCIONES DE MENÚ

En el sistema de requisiciones existen 6 opciones, las mismas que se activaran según el rol del usuario:

- Administración
- Inventario
- Requisición
- Chat
- Cuenta
- Salir

1. ADMINISTRACIÓN

El ingreso a esta opción solo lo realizara el usuario administrativo quien tiene acceso total a la gestión de usuarios.

1.4.4 Creación Cuenta de Usuario

Clic en Administración, donde se muestra la siguiente pantalla, se procederá a llenar los campos solicitados y clic en Grabar.

Administración del Sistema de Requisiciones

BENVENIDA, SECRETARIA VER EL SITIO / CAMBIAR CONTRASEÑA / TERMINAR SESIÓN

Inicio / Autenticación y autorización / Usuarios / Añadir usuario

Añadir usuario

Primero introduzca un nombre de usuario y una contraseña. Luego podrá editar el resto de opciones del usuario

Numero de cédula:

Únicamente dígitos, del grupo: 13 caracteres

Contraseña:

Su contraseña no puede comenzar con su información personal
Su contraseña debe contener al menos 8 caracteres
Su contraseña no puede ser una clave obvia común
Su contraseña no puede ser completamente numérica

Contraseña (confirmación):

Para verificar, introduzca la misma contraseña anterior

Ilustración 4: Creación Cuentas de Usuario

Una vez creada la cuenta se procede a ingresar los datos personales necesarios. Si la cuenta creada es un perfil de secretaria se activará la casilla *Gestión de Pedidos*, con la cual se le otorga los permisos para la gestión de inventario; además se activará la casilla *Es staff*

para otorgar a la cuenta los permisos para la gestión de usuarios. Si el perfil es operativo no se activará ninguna casilla. Al término, clic en Grabar.

Administración del Sistema de Requisiciones

BIENVENIDO/A. SECRETARIA. [VER EL SITIO](#) / [CAMBIAR CONTRASEÑA](#) / [TERMINAR SESIÓN](#)

[Inicio](#) > [Autenticación y autorización](#) > [Usuarios](#) > 0200427300

Modificar usuario

HISTÓRICO

Nombre de usuario:

0200427300

Requerido. 150 caracteres como máximo. Únicamente letras, dígitos y @/./+/-/_.

Contraseña:

algoritmo: pbkdf2_sha256 iteraciones: 36000 sal: PgspG4***** función resumen: 2lz5FQ*****

Las contraseñas no se almacenan en texto plano, por lo que no hay forma de ver la de este usuario, pero puede cambiar la contraseña usando [este formulario](#).

Información personal

Nombre:

Secretaria

Apellidos:

Oficina Principal

Dirección de correo electrónico:

angelicagarcia2017@gmail.com

Permisos

☒ Activo

Indica si el usuario debe ser tratado como activo. Desmarque esta opción en lugar de borrar la cuenta.

☒ Es staff

Indica si el usuario puede entrar en este sitio de administración.

PERFIL

Perfil: Secretaria Oficina Principal

Fecha de nacimiento:

28/03/1948

Hoy |

☐ Gestión de pedidos

Imagen de perfil:

Seleccionar archivo

No se eligió archivo

Eliminar

Grabar y añadir otro

Grabar y continuar editando

GRABAR

Ilustración 5: Asignación de Roles

2. INVENTARIO

El ingreso a esta opción solo lo realizara el usuario administrativo quien tiene a cargo el ingreso de las categorías y productos al inventario.



Ilustración 6: Opción Inventario

1.5

1.6 CATEGORIAS

1.6.1 Ingreso Listado Categorías

Al ingresar en el menú Inventario/Categorías, se observa un listado de las categorías ya existentes.

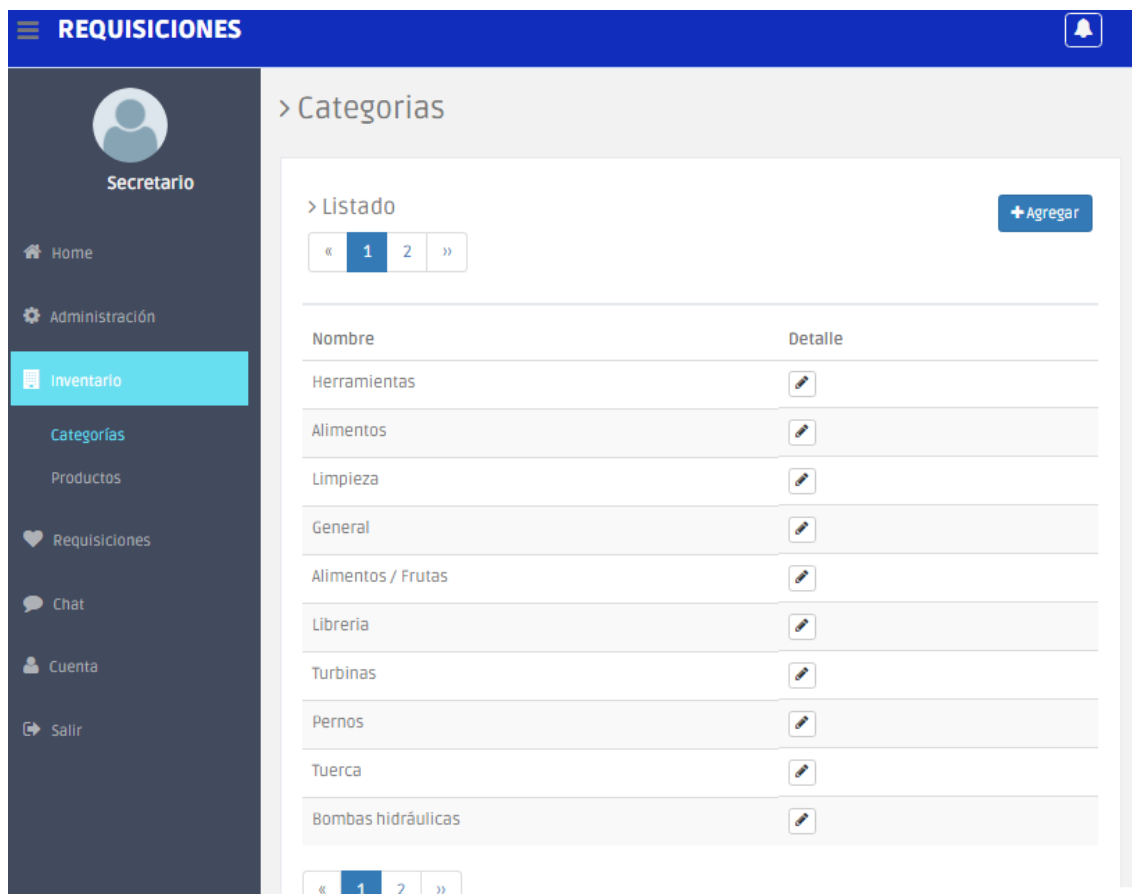


Ilustración 7: Listado de Categorías

1.6.2 Creación de Nueva Categoría

Clic en Agregar, ingresamos la información necesaria como *Nombre* de la categoría y *Padre* que se refiere a la subcategoría que pertenece. Clic en Guardar.

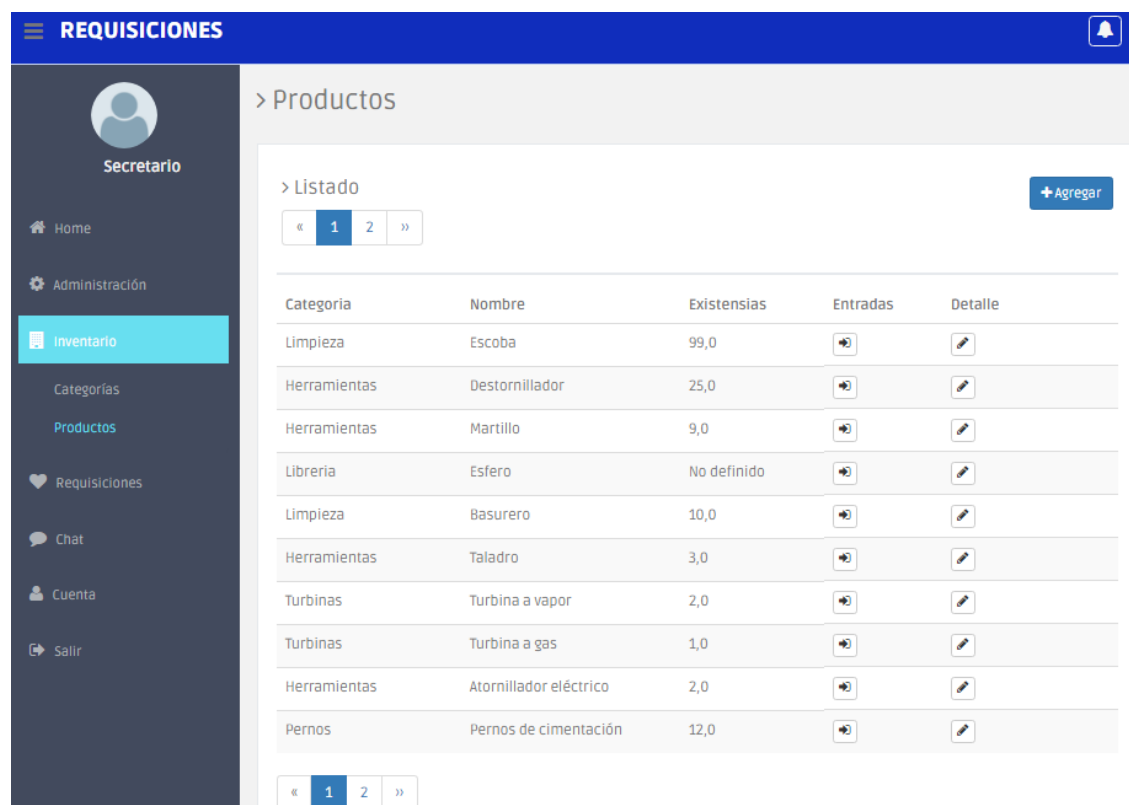


Ilustración 8: Creación de Categorías

1.7 PRODUCTOS

1.7.1 Ingreso Listado Productos

Al ingresar en el menú Inventario/Productos, se observa un listado de los productos ingresados.



Categoría	Nombre	Existencias	Entradas	Detalle
Limpieza	Escoba	99,0		
Herramientas	Destornillador	25,0		
Herramientas	Martillo	9,0		
Librería	Esfero	No definido		
Limpieza	Basurero	10,0		
Herramientas	Taladro	3,0		
Turbinas	Turbina a vapor	2,0		
Turbinas	Turbina a gas	1,0		
Herramientas	Atornillador eléctrico	2,0		
Pernos	Pernos de cimentación	12,0		

Ilustración 9: Listado de Productos

1.7.2 Creación Nuevo Producto

Clic en Agregar, ingresamos la información necesaria como *Nombre* de producto y categoría o subcategoría a la cual pertenece el producto. Clic en Guardar.

REQUISICIONES

Secretario

Home

Administración

Inventario

Categorías

Productos

Requisiciones

Chat

Cuenta

> Productos

> Datos de producto

Nombre: Eléctricas

Categoría: Bombas hidráulicas / Tensionador de pernos

Guardado

Regresar

Herramientas

Alimentos

Limpieza

General

Alimentos / Frutas

Librería

Turbinas

Pernos

Tuerca

Bombas hidráulicas

Bombas hidráulicas / Tensionador de pernos

Ilustración 10: Creación Nuevo Producto

Una vez ingresado el nombre del producto con su respectiva categoría. Este aparecerá en el listado de productos.

Bombas hidráulicas / Tensionador de pernos	Eléctricas	No definido		
--	------------	-------------	--	--

Ilustración 11: Producto Nuevo Ingresado

1.7.3 Ingreso o Editar de Datos del Producto

Ingresar a Inventario / Productos

Clic en ingresa los datos del producto. Clic en Guardar. Caso contrario Clic en Regresar.

REQUISICIONES

Secretario

Home

Administración

Inventario

Categorías

Productos

Requisiciones

Chat

> Productos


> Entrada a stock de Bombas hidráulicas / Tensionador de pernos > Eléctricas

Cantidad

Nota

Guardar Regresar


Ilustración 12: Detalle del Producto


Clic en  editar datos del producto. Clic en Guardar. Caso contrario Clic en Regresar.

1.7.4

1.7.5 Eliminar producto

Ingresar a Inventario / Productos

Clic en  .

Clic  si se desea eliminar el producto.

REQUISICIONES

Secretario

Home

Administración

Inventario

Categorías

Productos

> Productos

> Datos de producto

Nombre Eléctricas

Categoría Bombas hidráulicas / Tensionador de pernos

Guardar Regresar

Ilustración 13: Eliminar Producto

3. REQUISICIONES

El ingreso a esta opción lo realiza el usuario administrativo y operativo, ambos roles pueden generar un pedido de requisición, además el perfil administrativo podrá gestionar lugares y generar reportes.

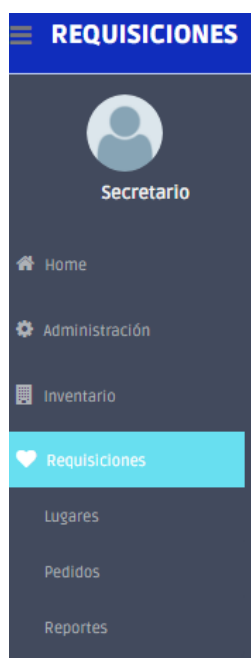


Ilustración 14: Perfil Administrativo

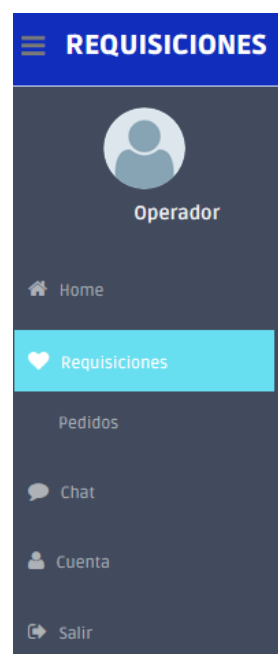


Ilustración 15: Perfil Operativo

1.7.6

1.8 LUGARES

1.8.1 Ingreso Listado Lugares

Clic Requisiciones / Lugares

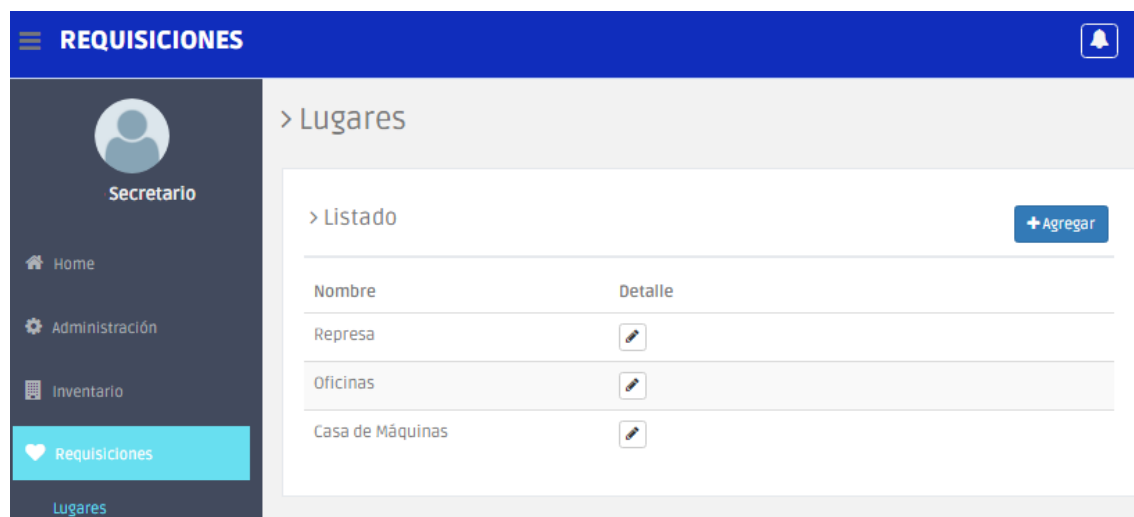
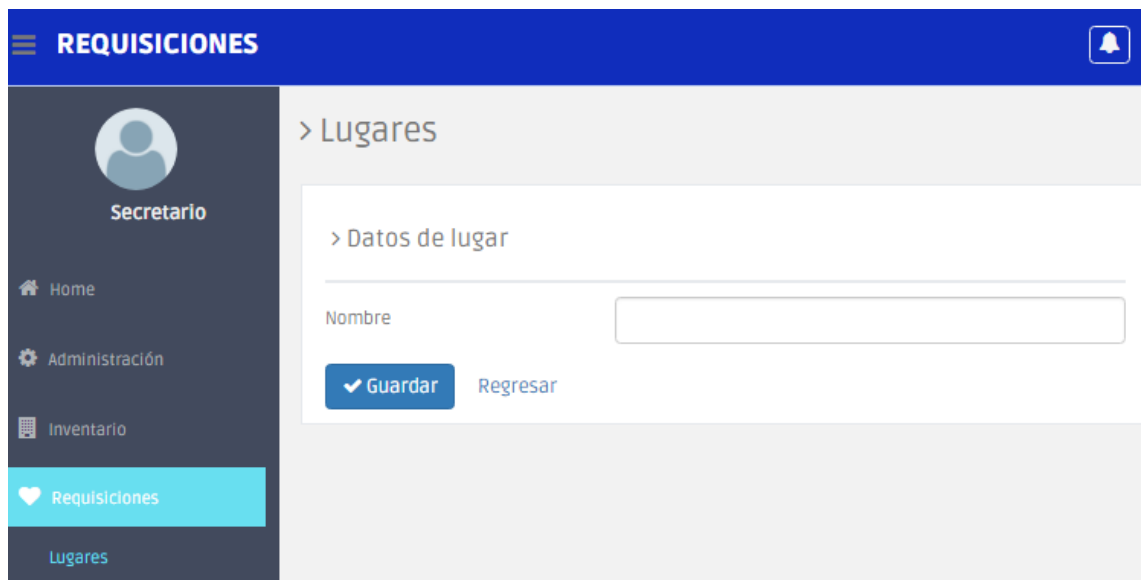


Ilustración 16: Listado de Lugares

1.8.2 Creación Lugar

Clic en Agregar, ingresamos el *Nombre* del lugar. Clic en Guardar.




The screenshot shows a web application interface for 'REQUISICIONES'. On the left is a dark sidebar with a user profile 'Secretario' and a menu with options: Home, Administración, Inventario, Requisiciones (highlighted in light blue), and Lugares. The main content area has a blue header with the title 'REQUISICIONES' and a notification bell icon. Below the header, the page title is '> Lugares'. Underneath, there's a section titled '> Datos de lugar' containing a text input field labeled 'Nombre'. At the bottom of this section are two buttons: a blue '✓ Guardar' button and a grey 'Regresar' button.

Ilustración 17: Creación Lugar

1.8.3

1.8.4 Eliminar Lugar

Ingresar a Requisiciones / Lugar

Clic en  .


Clic  si se desea eliminar el lugar.



Ilustración 18: Eliminar Lugar

1.9 PEDIDOS

1.9.1 Creación de Pedido

Al ingresar en el menú Requisiciones/Pedidos, se observa un formulario donde se ingresa la cantidad del producto seleccionado previamente de un listado, además se agregará alguna nota descriptiva del pedido de requisición. Al terminar, Clic en Generar o Guardar según desee.

REQUISICIONES

Operador

Home

Requisiciones

Pedidos

Chat

Cuenta

Salir

> Pedidos

> Lineas de pedido

Cantidad	Producto	Eliminar?
2	Turbinas > Turbina a gas	<input type="checkbox"/>
1	Herramientas > Taladro	<input type="checkbox"/>
	-----	<input type="checkbox"/>

+ Agregar linea

Entregar en: Represa

Nota general

Guardar Generar Regresar

Ilustración 19: Creación Pedido

Al ser generada la petición es enviada una notificación a la *Secretaria* para su posterior revisión se muestra en la imagen representada por una campaña.

1.9.2 Procesamiento del Pedido

Una vez notificada la secretaria, clic en la notificación o a su vez ingresando *Requisiciones/Pedidos*. Clic muestra las siguientes opciones:

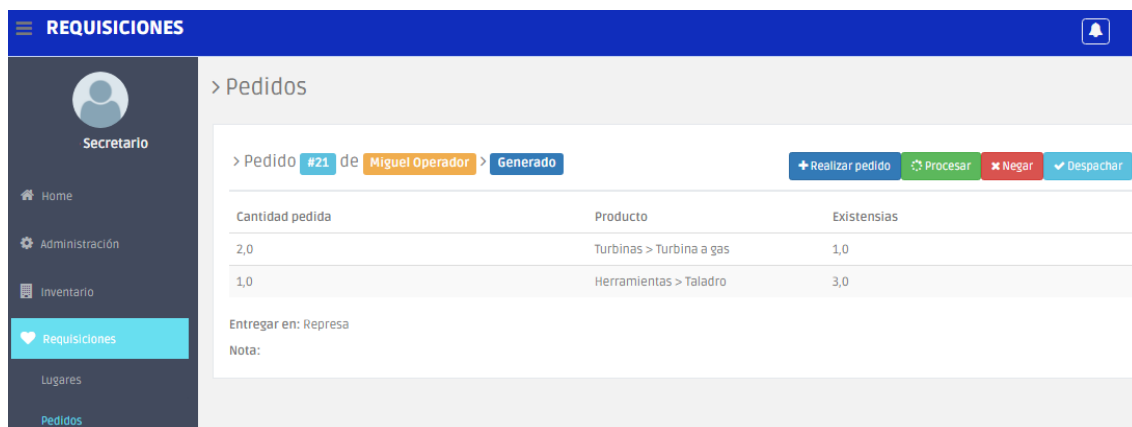


Ilustración 20: Procesamiento Pedido


- **Procesar** Aceptar el pedido de requisición por parte de secretaria para su posterior despacho. Al presionar este botón se muestra la siguiente pantalla.

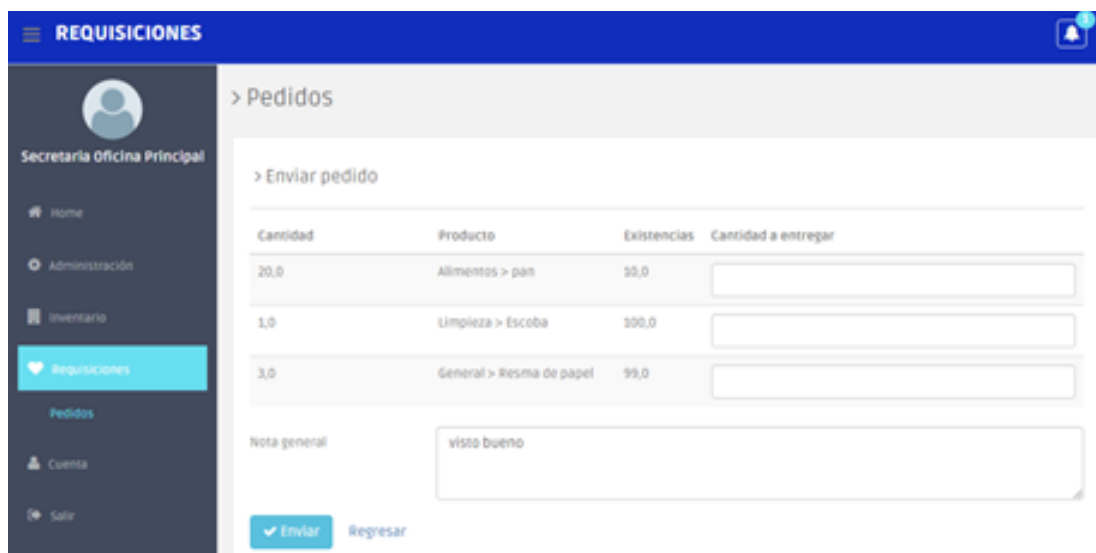
Ilustración 21: Botón Procesar

- **Negar** notificación rechazo del pedido de requisición por políticas de la empresa o toma de decisiones, responsabilidad de la secretaria.

Ilustración 22: Botón Negar

- **Cancelar pedido** notificación de cancelar el pedido de requisición por políticas de la empresa o toma de decisiones, responsabilidad de la secretaria.

-  Notificación de despacho hacia el lugar donde se realizó el pedido de requisición. Al presionar este botón, se muestra la siguiente pantalla donde se despacha el pedido según el stock de existencia en inventario. Clic *Enviar*.




Cantidad	Producto	Existencias	Cantidad a entregar
20,0	Alimentos > pan	30,0	<input type="text"/>
1,0	Limpieza > Escoba	100,0	<input type="text"/>
3,0	General > Resma de papel	99,0	<input type="text"/>

Nota general: visto bueno

Ilustración 23: Botón Despachar

La siguiente opción, la tiene disponible el perfil del operador, una vez que la Secretaria haya Despachado el pedido.

-  Notificación que el operador envía cuando haya recibido el pedido de requisición.

1.10 REPORTES

1.10.1 Creación de Reportes

Al ingresar en el menú Requisiciones / Reportes, se podrá generar reportes en base a 3 opciones (estado, fecha, empleado).

1.10.2 Reporte por estado

- Clic Requisiciones / Reportes
- Escoger la opción **Por estado**
- Escoger el estado
- Clic Generar

REPORTE DE PEDIDOS POR ESTADO

Estado: Recibido

Código	Usuario	Fecha	Estado	Lugar
20	Miguel Operador	2018-02-21 12:37:20.849614+00:00	Recibido	Casa de Máquinas
19	Patricio Orozco	2018-02-04 18:08:51.436387+00:00	Recibido	Represa
11	Patricio Orozco	2017-12-20 20:05:50.730615+00:00	Recibido	Casa de Máquinas
8	Patricio Orozco	2017-12-13 05:17:09.240214+00:00	Recibido	Casa de Máquinas
5	Patricio Orozco	2017-12-12 22:02:01.581888+00:00	Recibido	Casa de Máquinas

Ilustración 24: Reporte por estado

1.10.3 Reporte por fecha

- Clic Requisiciones / Reportes
- Escoger la opción **Por fecha**
- Escoger el rango de fechas para el reporte
- Clic Generar

REPORTE DE PEDIDOS POR RANGO DE FECHAS

Desde: 2018-01-01 Hasta: 2018-02-01

Código	Usuario	Fecha	Estado	Lugar
18	Diana Gomez	2018-01-29 21:00:22.321382+00:00	Despachado	Represa
17	Patricio Orozco	2018-01-24 12:24:39.102930+00:00	Negado	Casa de Máquinas
16	Patricio Orozco	2018-01-16 18:11:15.404864+00:00	Generado	Oficinas
15	Patricio Orozco	2018-01-15 19:39:43.769329+00:00	En proceso	Represa

Ilustración 25: Reporte por fechas

1.10.4 Reporte por empleado

- Clic Requisiciones / Reportes
- Escoger la opción **Por empleado**
- Escoger el nombre del empleado.
- Clic Generar

REPORTE DE PEDIDOS POR USUARIO

Usuario: Angélica García

Código	Usuario	Fecha	Estado	Lugar
10	Angélica García	2017-12-20 19:47:43.640511+00:00	Cancelado	Casa de Máquinas
9	Angélica García	2017-12-20 19:18:40.183137+00:00	Borrador	Casa de Máquinas

Ilustración 26: Reporte por empleado

4. CHAT

El ingreso a esta opción lo realiza el usuario administrativo y operativo, ambos tienen acceso al chat.

Se muestra el listado de usuarios registrados en el sistema, para empezar un chat deberá escoger el usuario, ingresar el mensaje y clic Enviar

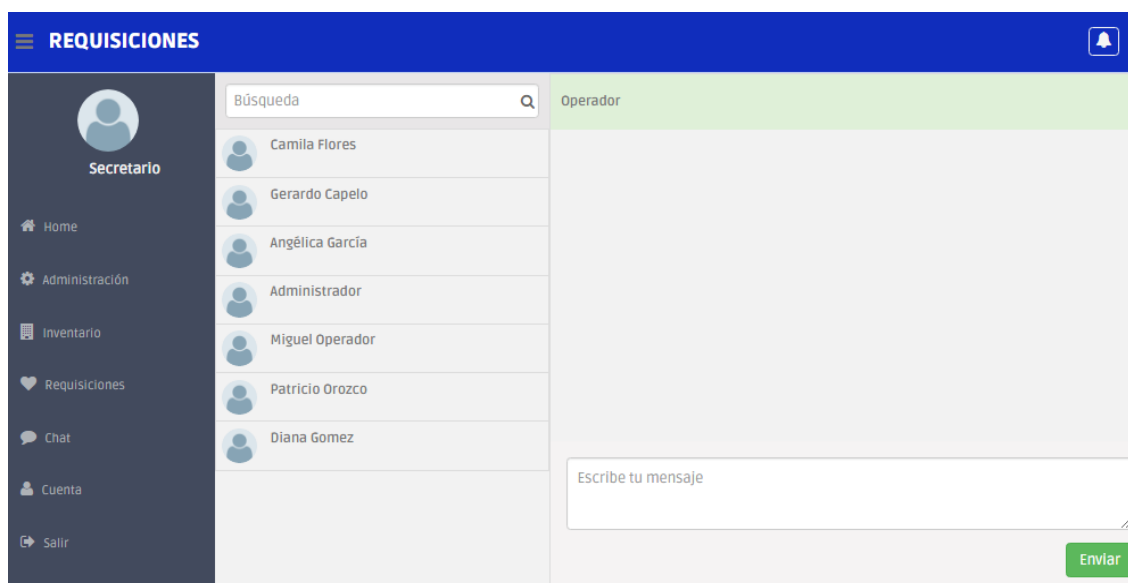


Ilustración 27: Opción Chat

5. CUENTA

El ingreso a esta opción lo realiza el usuario administrativo y operativo, ambos tienen acceso a su perfil y al cambio de contraseña.

1.10.5 Acceso a perfil de usuario

Clic Cuenta / Mi perfil

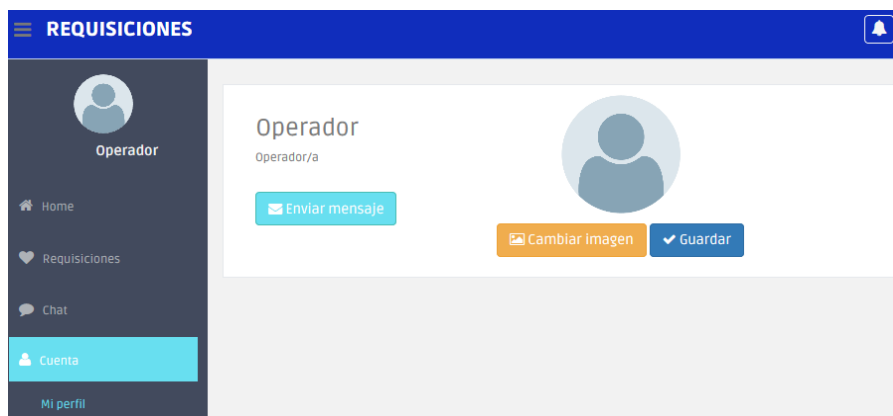


Ilustración 28: Mi perfil

1.10.6 Cambio de Contraseña

- Clic Cuenta / Cambiar contraseña
- Relleno los datos sugeridos.
- Clic Guardar.

Ilustración 29: Cambio de Contraseña

6. SALIR

Esta opción se ejecutará cuando el usuario desee salir del sistema de requisiciones.



Ilustración 30: Opción Salir



ANEXO E
ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
INGENIERÍA EN SISTEMAS

SUS (SYSTEMS USABILITY SCALE)
TEST DE USABILIDAD
SISTEMA DE REQUISICIONES



TIPO DE USUARIO: ADMINISTRATIVO ☐
OPERATIVO ☐

1. CREO QUE ME GUSTARÍA UTILIZAR ESTE SISTEMA FRECUENTEMENTE

1	2	3	4	5
MUY EN DESACUERDO				TOTALMENTE DE ACUERDO

2. EL SISTEMA ME RESULTÓ INNECESARIAMENTE COMPLEJO.

1	2	3	4	5
MUY EN DESACUERDO				TOTALMENTE DE ACUERDO

3. CREO QUE EL SISTEMA ES BASTANTE FÁCIL DE UTILIZAR.

1	2	3	4	5
MUY EN DESACUERDO				TOTALMENTE DE ACUERDO

4. CREO QUE NECESITARÍA EL SOPORTE DE UN TÉCNICO PARA PODER UTILIZAR EL SISTEMA.

1	2	3	4	5
MUY EN DESACUERDO				TOTALMENTE DE ACUERDO

5. CREO QUE LAS DIFERENTES FUNCIONES DEL SISTEMA SE ENCUENTRAN MUY BIEN INTEGRADAS.

1	2	3	4	5
MUY EN DESACUERDO				TOTALMENTE DE ACUERDO

6. OPINO QUE HUBO DEMASIADA INCONSISTENCIA EN EL SISTEMA.

1	2	3	4	5
MUY EN DESACUERDO				TOTALMENTE DE ACUERDO

7. IMAGINO QUE LA MAYORÍA DE LAS PERSONAS APRENDERÍA A UTILIZAR EL SISTEMA RÁPIDAMENTE.

1	2	3	4	5
MUY EN DESACUERDO				TOTALMENTE DE ACUERDO

8. ME SENTÍ ALGO INCÓMODO AL UTILIZAR ESTE SISTEMA.

1	2	3	4	5
MUY EN DESACUERDO				TOTALMENTE DE ACUERDO

9. ME SENTÍ MUY SEGURO AL UTILIZAR ESTE SISTEMA.

1	2	3	4	5
MUY EN DESACUERDO				TOTALMENTE DE ACUERDO

10. NECESITO APRENDER MUCHAS OTRAS COSAS ANTES DE PODER UTILIZAR CORRECTAMENTE EL SISTEMA.

1	2	3	4	5
MUY EN DESACUERDO				TOTALMENTE DE ACUERDO